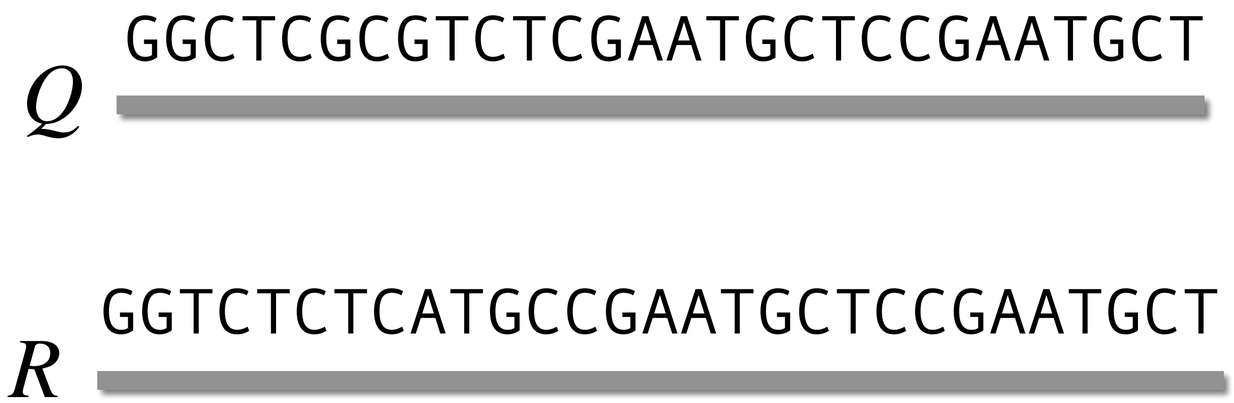
A distance-based likelihood framework for genome-wide (phylogenetic) pattern matching

Ali Osman Berk Şapcı & Siavash Mirarab
UC San Diego

RECOMB-Seq 2026 **[extended]**

My promise for this talk

Given a query genome and a reference, and a distance Δ

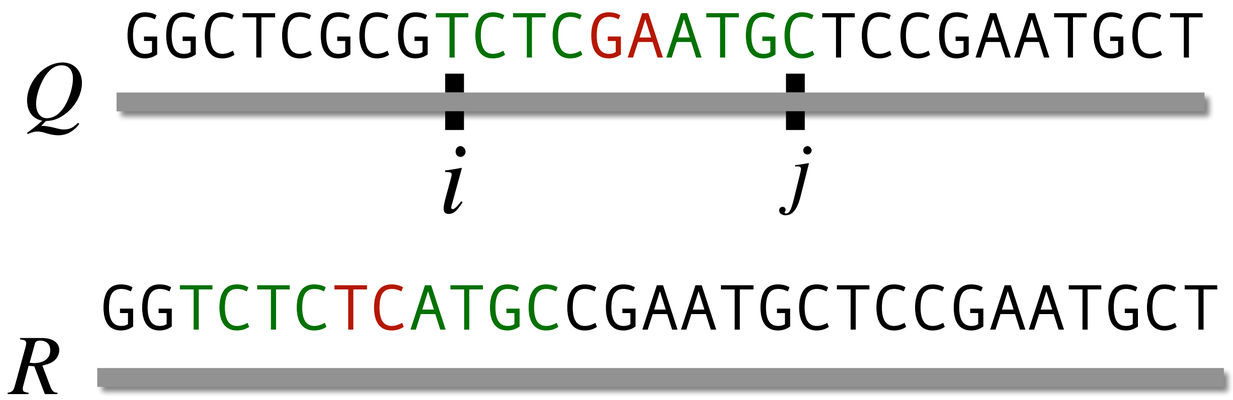


My promise for this talk

Given a query genome and a reference, and a distance Δ

I) Constant time queries for a given segment $[i,j)$ of Q:

Is there a *(quasi)-homologous* counterpart in R with distance $< \Delta$ to $Q_{i:j}$



2 mismatches:

$\Delta = 0.25$ ✓

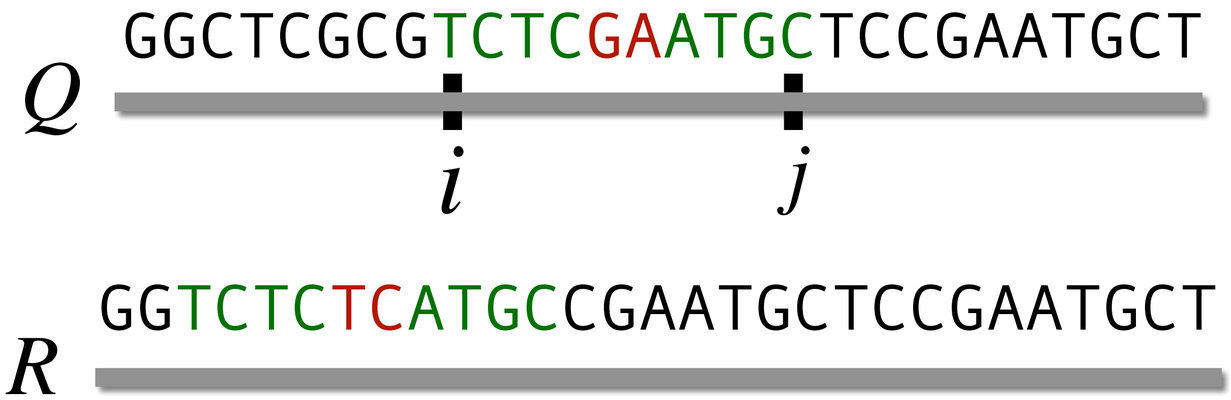
$\Delta = 0.1$ ✗

My promise for this talk

Given a query genome and a reference, and a distance Δ

I) Constant time queries for a given segment $[i,j)$ of Q:

Is there a *(quasi)-homologous* counterpart in R with distance $< \Delta$ to $Q_{i:j}$



2 mismatches:

$\Delta = 0.25$ ✓

$\Delta = 0.1$ ✗

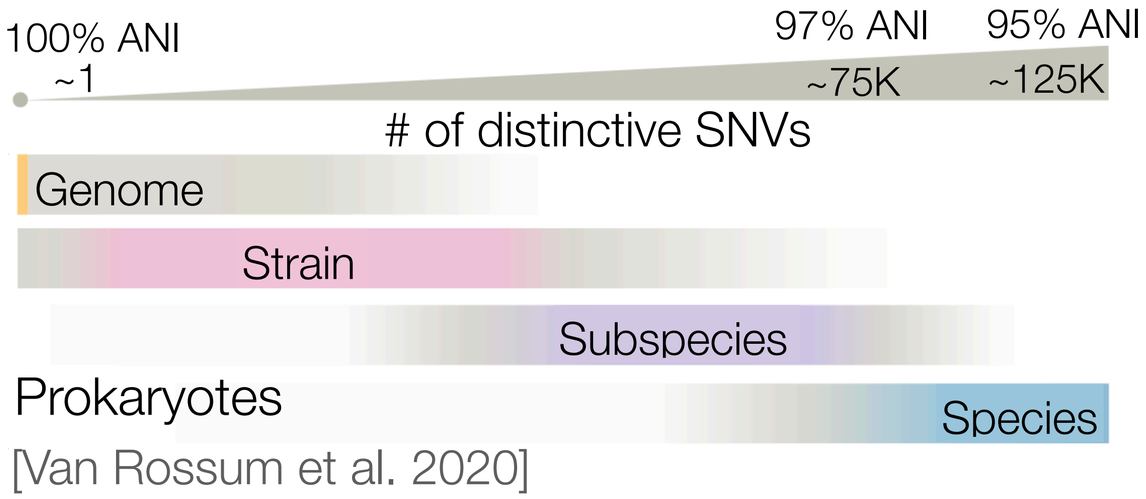
II) Linear time enumeration of all maximal segments of Q:

with a *(quasi)-homologous* counterpart in R with distance $< \Delta$ (or $> \Delta$)

Genomes are evolutinoarily heterogeneous

Genome-wide average nucleotide identity (ANI):

- comparing genomes and MAGs
- defining species and taxa (e.g., GTDB)

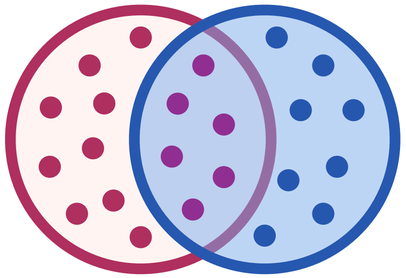


[Van Rossum et al. 2020]

Genomes are evolutinoarily heterogeneous

Genome-wide average nucleotide identity (ANI):

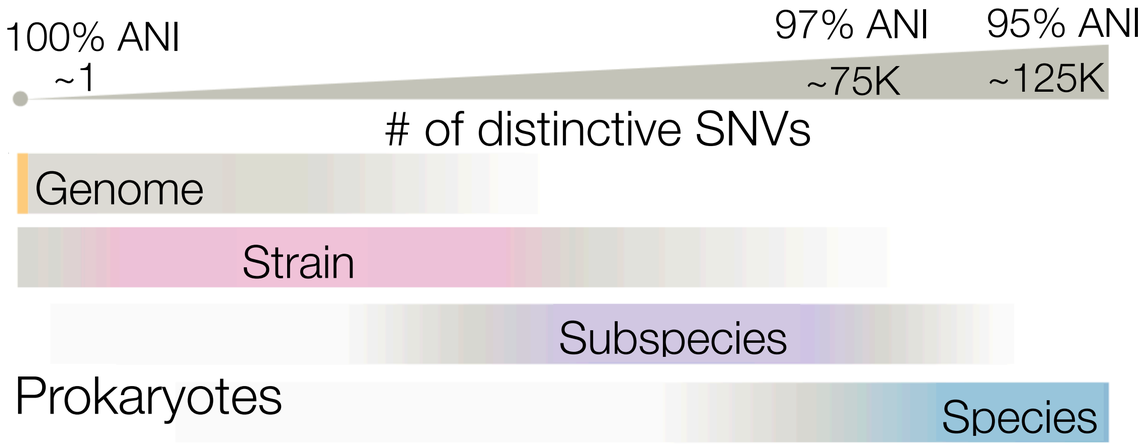
- comparing genomes and MAGs
- defining species and taxa (e.g., GTDB)



Jaccard index of *k*-mers sets

$$D = 1 - \frac{2J}{J+1}^{1/k}$$

Many flavors:
Mash, skani for MAGs, Skmer for genome skims
FastANI & orthoANI

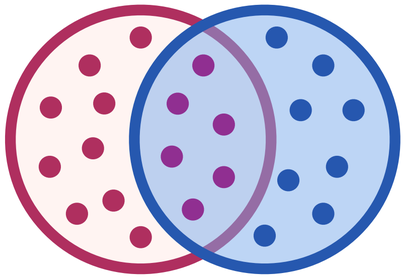


[Van Rossum et al. 2020]

Genomes are evolutinoarily heterogeneous

Genome-wide average nucleotide identity (ANI):

- comparing genomes and MAGs
- defining species and taxa (e.g., GTDB)

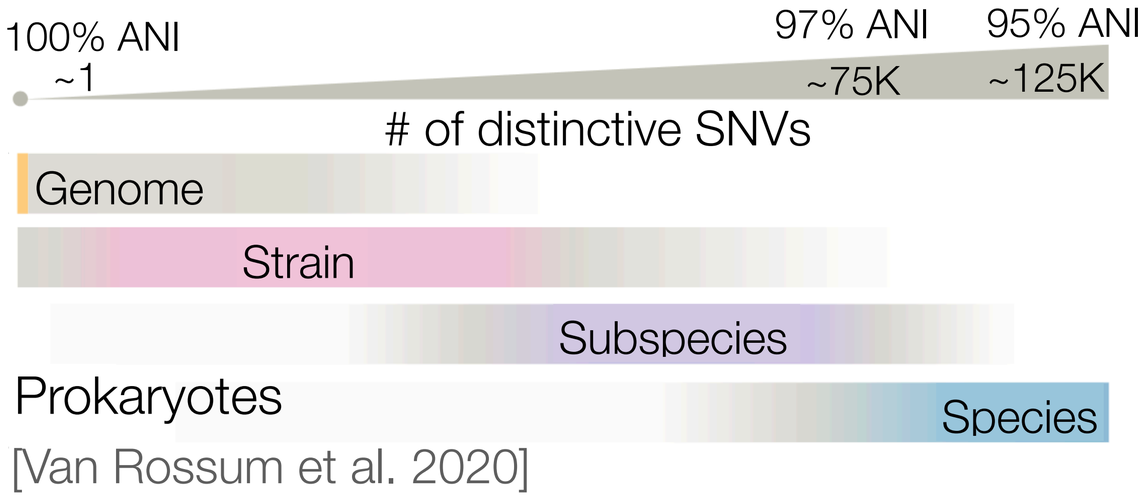


Jaccard index of *k*-mers sets

$$D = 1 - \frac{2J}{J+1}^{1/k}$$

Many flavors:
Mash, skani for MAGs, Skmer for genome skims
FastANI & orthoANI

ANI is just a summary statistic…



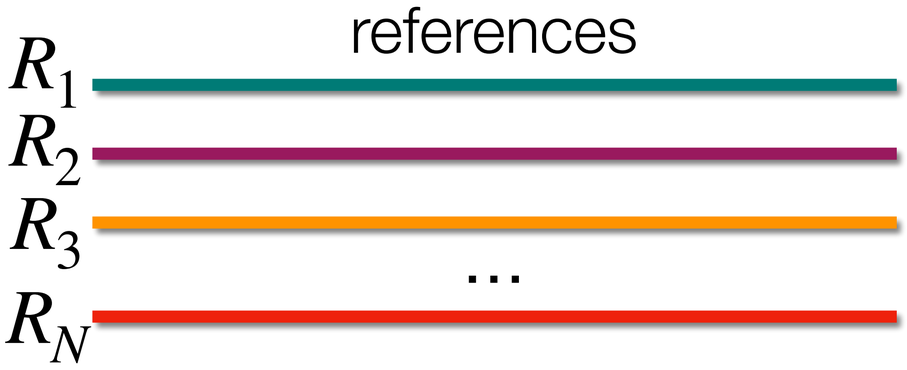
[Van Rossum et al. 2020]

Causes for local deviations:

- Horizontally transferred genes
- Conserved elements
- Contamination in assemblies
- Viral integration

Neutral scenario: change in distances due to rate variation

query genome Q

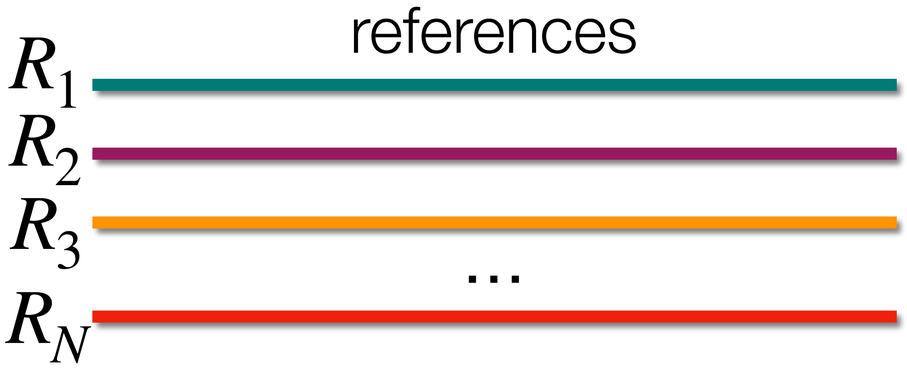


Neutral scenario: change in distances due to rate variation

query genome

sliding window

find the best matching window in R

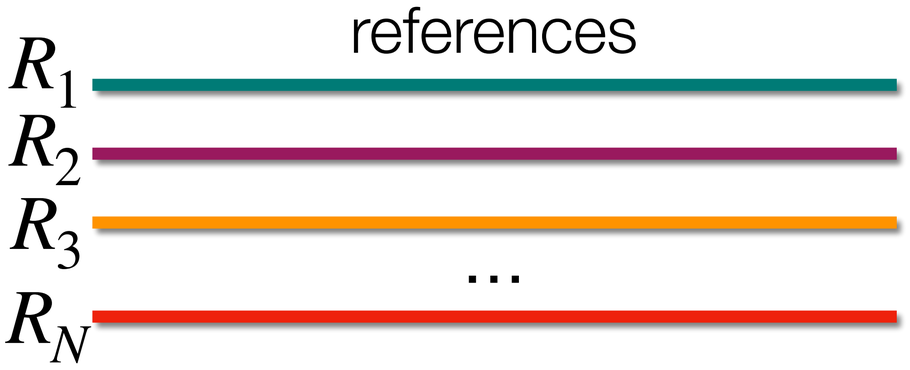


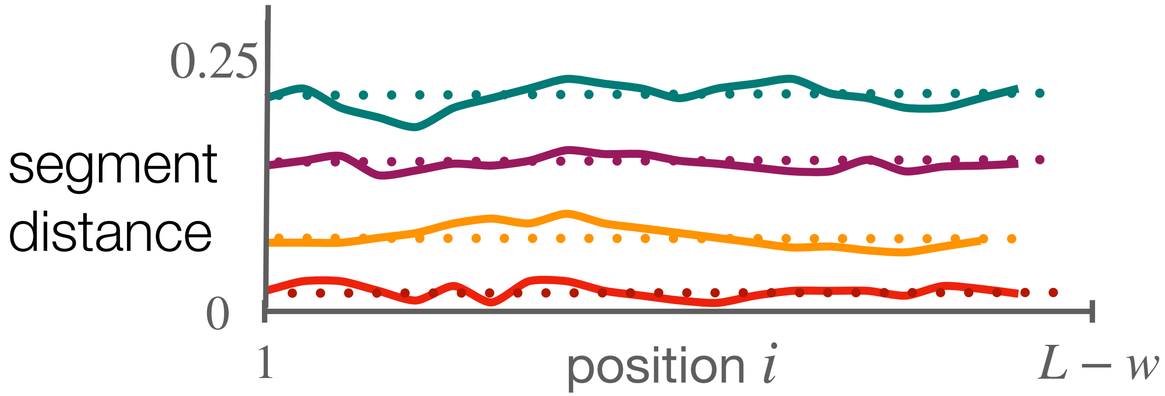
Neutral scenario: change in distances due to rate variation

query genome

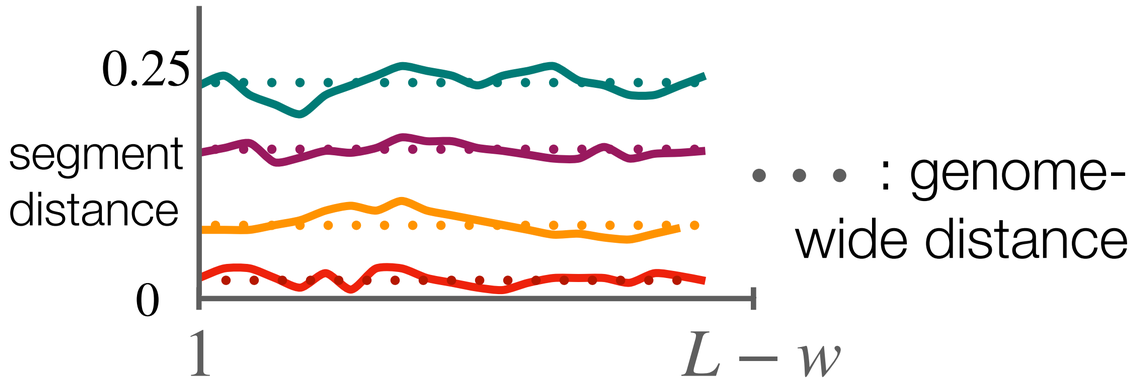
sliding window

find the best matching window in R



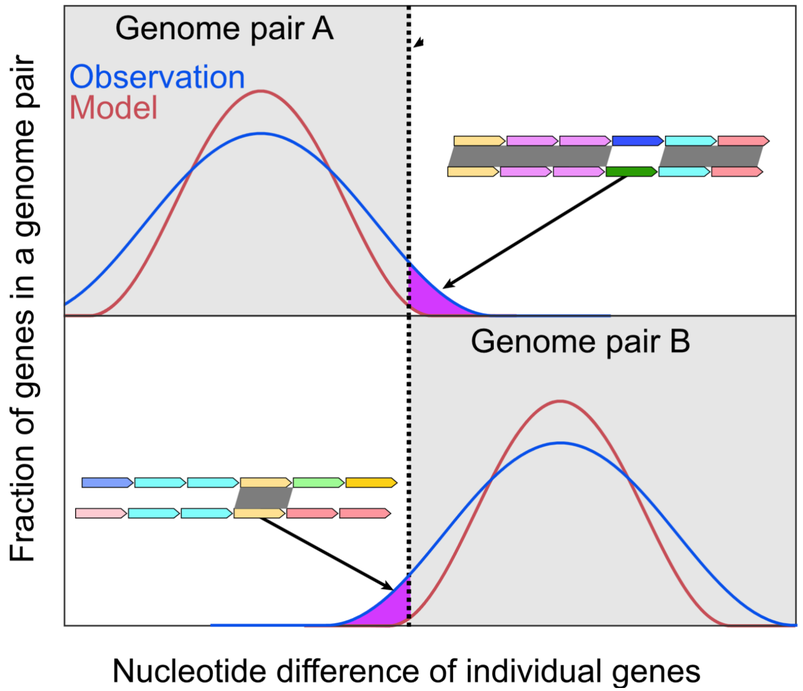


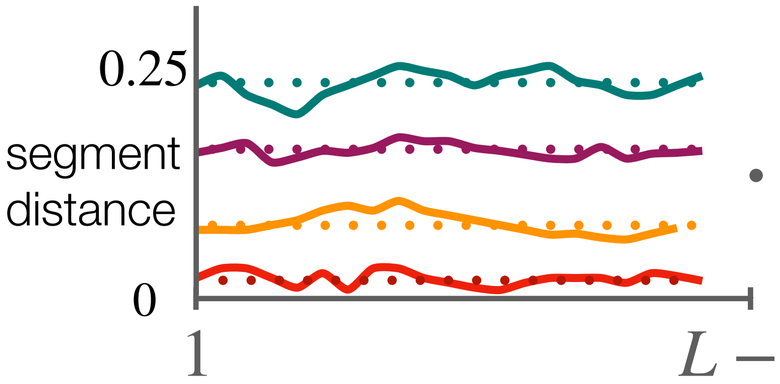
· · · · · · · : genome-wide distance



• • • : genome-wide distance

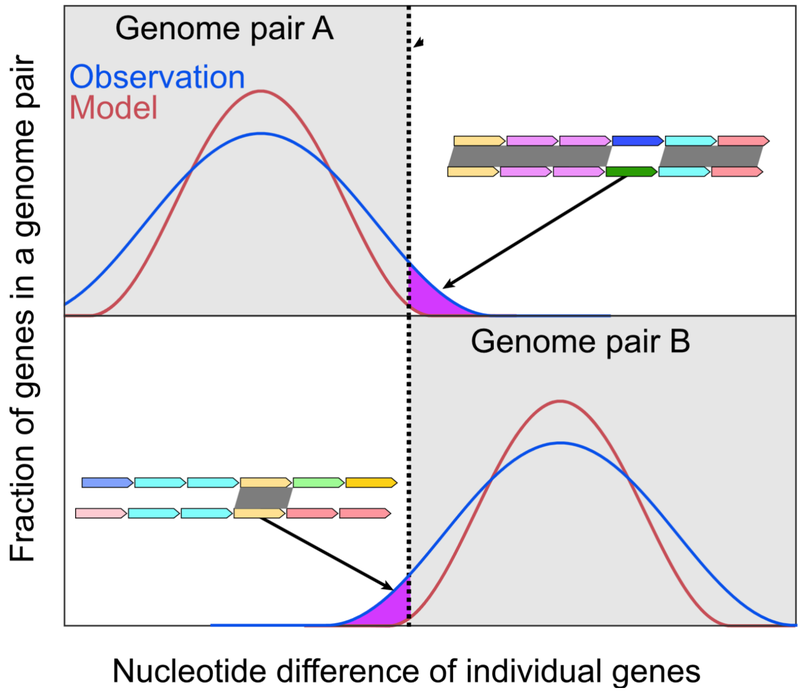
e.g., horizontal gene transfer



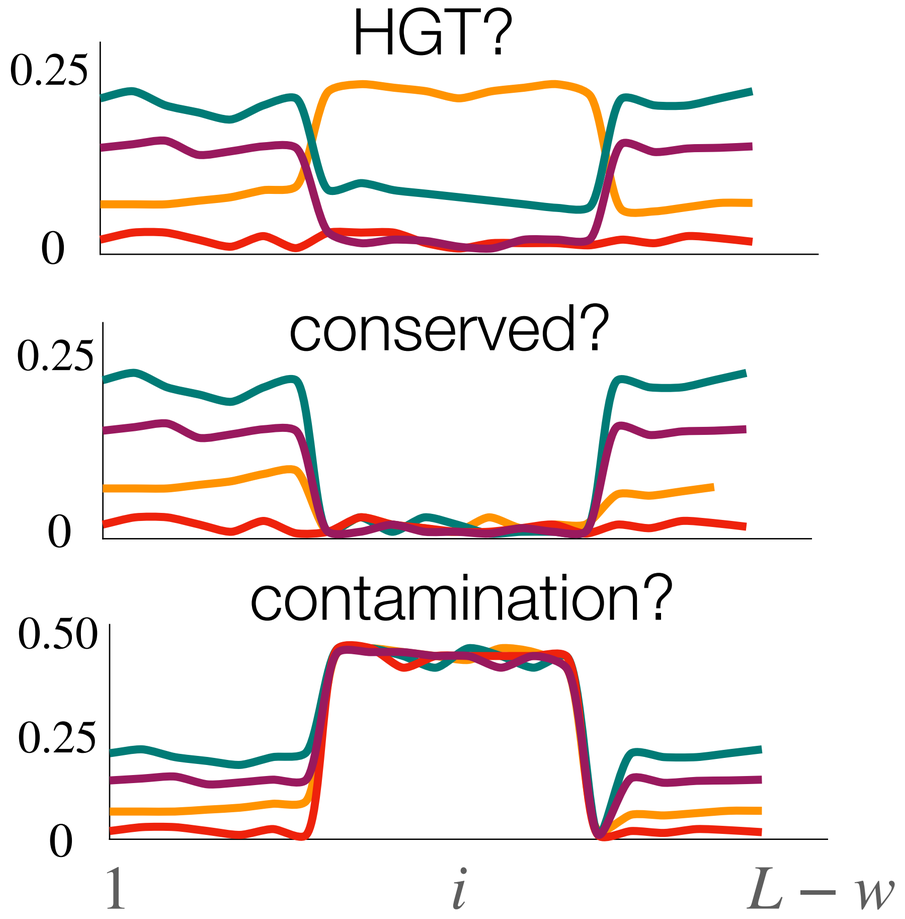


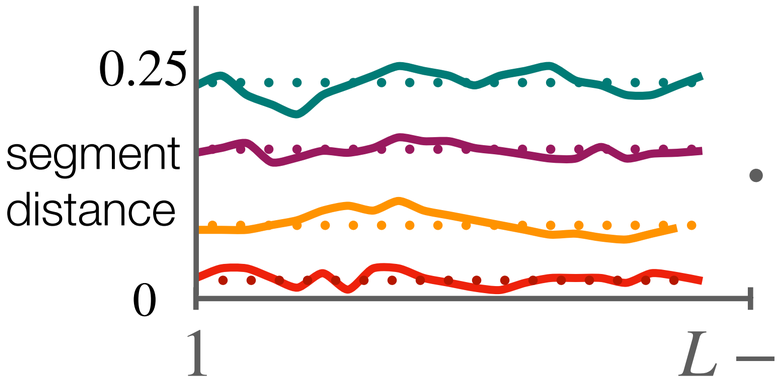
• • • : genome-wide distance

e.g., horizontal gene transfer



segments w/ deviation

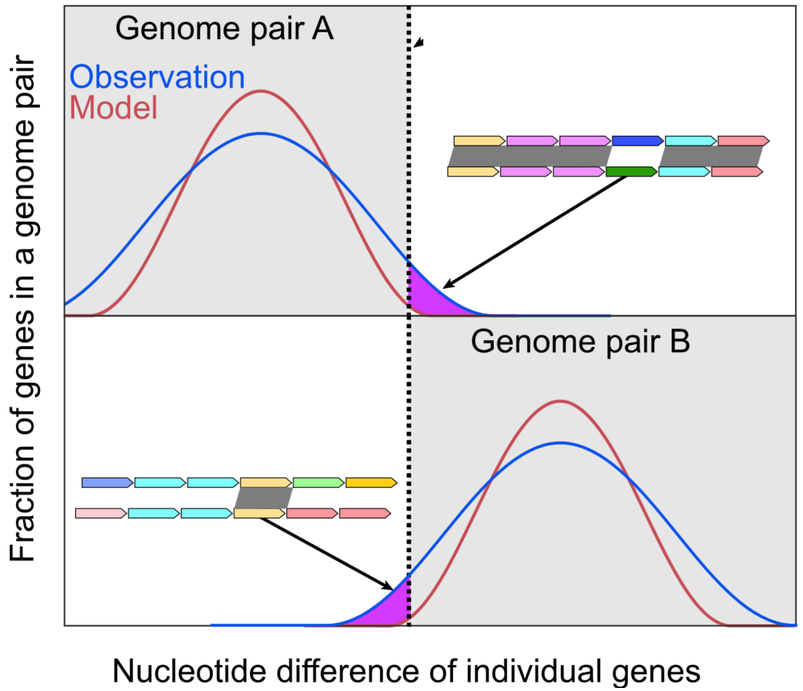




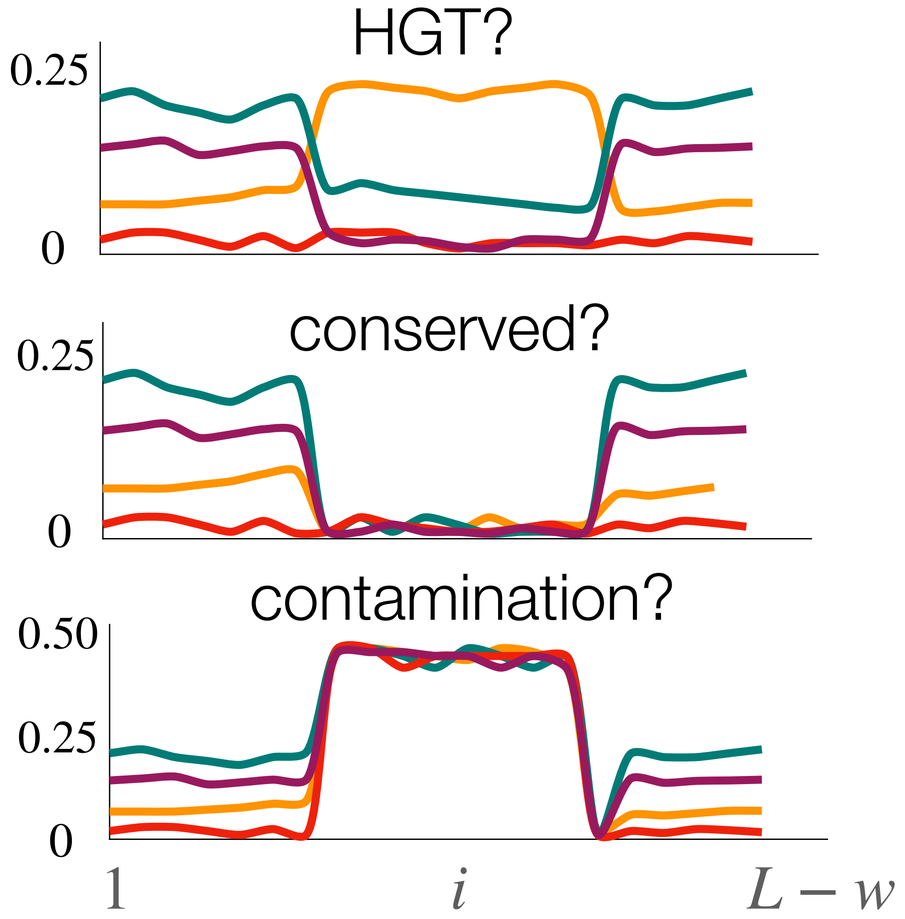
• • • : genome-wide distance

Goal: → measure local distances,
→ detect deviations and outlier regions

e.g., horizontal gene transfer



segments w/ deviation



Ideally, no fixed window size: distance of any segment

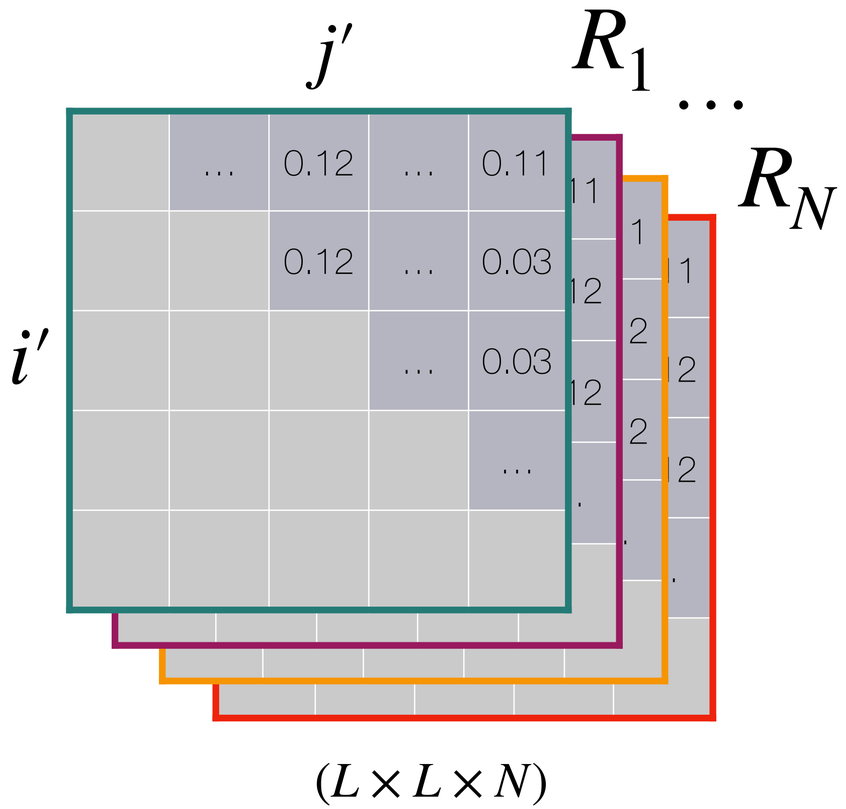
There is usually a more elegant solution than sliding windows.

query genome Q

For every reference, $R \in \mathscr{R}$ compute:

- segment distances $\mathrm{d}(Q_{i:j}, R_{i':j'})$
- genome-wide distances $\mathrm{ANI}(Q, R)$

Then, **jointly compare** for $\mathscr{R}$…



Ideally, no fixed window size: distance of any segment

There is usually a more elegant solution than sliding windows.

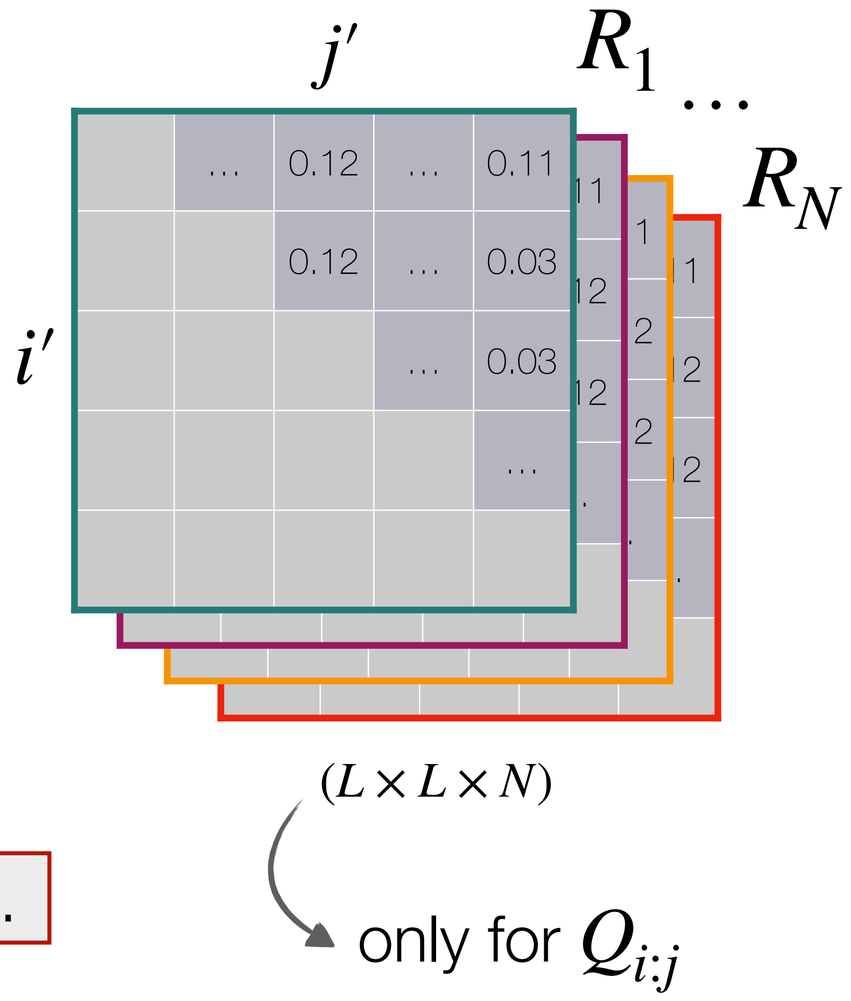
query genome Q

For every reference, $R \in \mathscr{R}$ compute:

- segment distances $\mathrm{d}(Q_{i:j}, R_{i':j'})$
- genome-wide distances $\mathrm{ANI}(Q, R)$

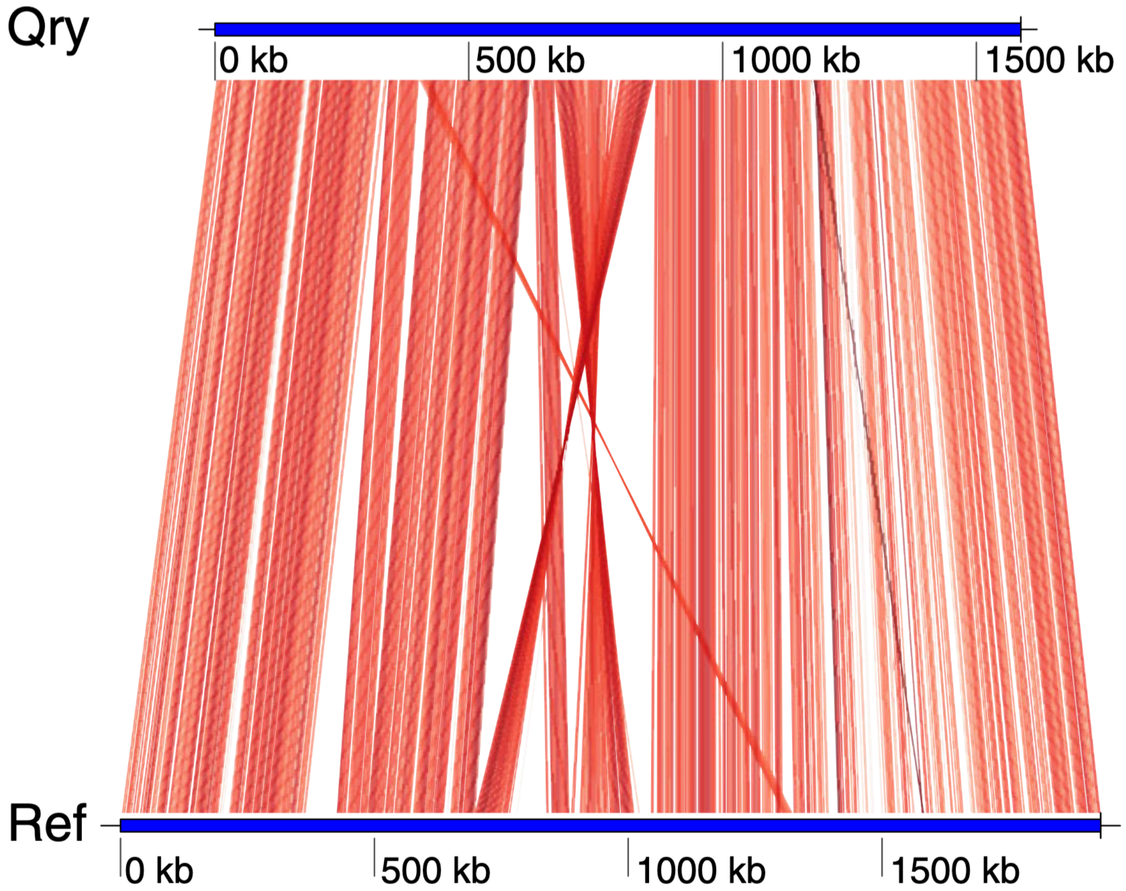
Then, **jointly compare** for $\mathscr{R}$…

One might as well do WGA…



What about homology mapping?

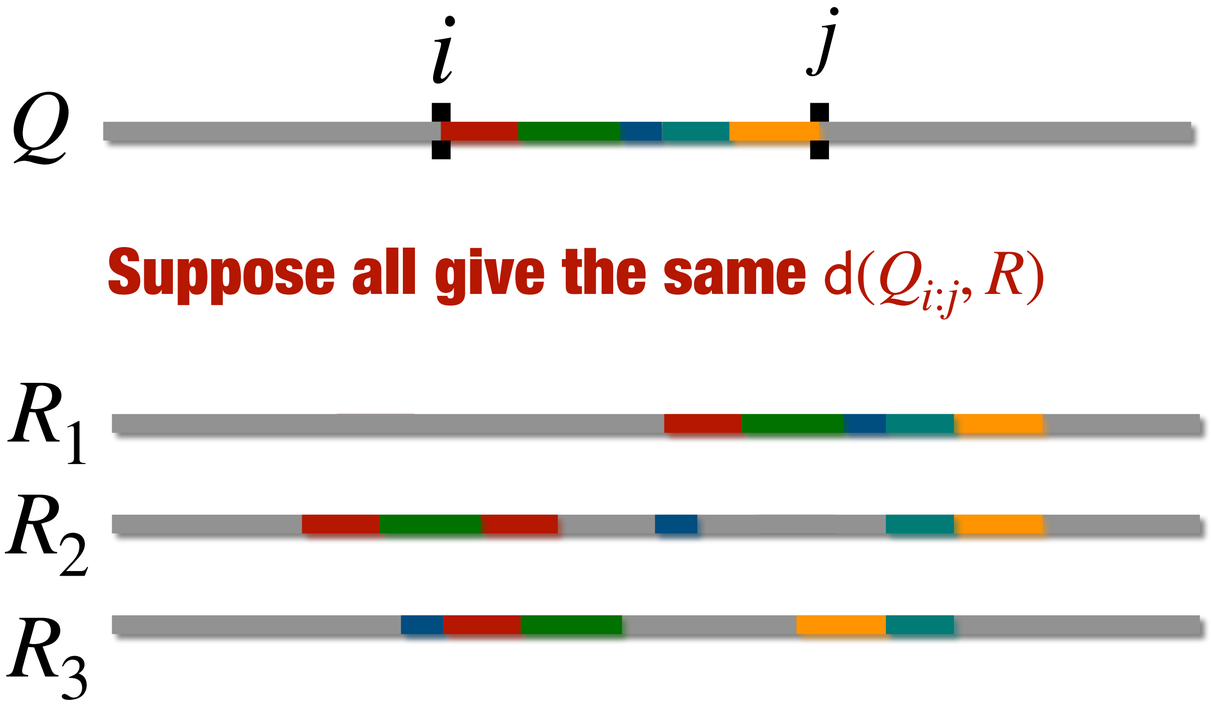
Window-based homology mapping using MashMap



FastANI [Jain et al. 2018]

Our focus is outlier regions

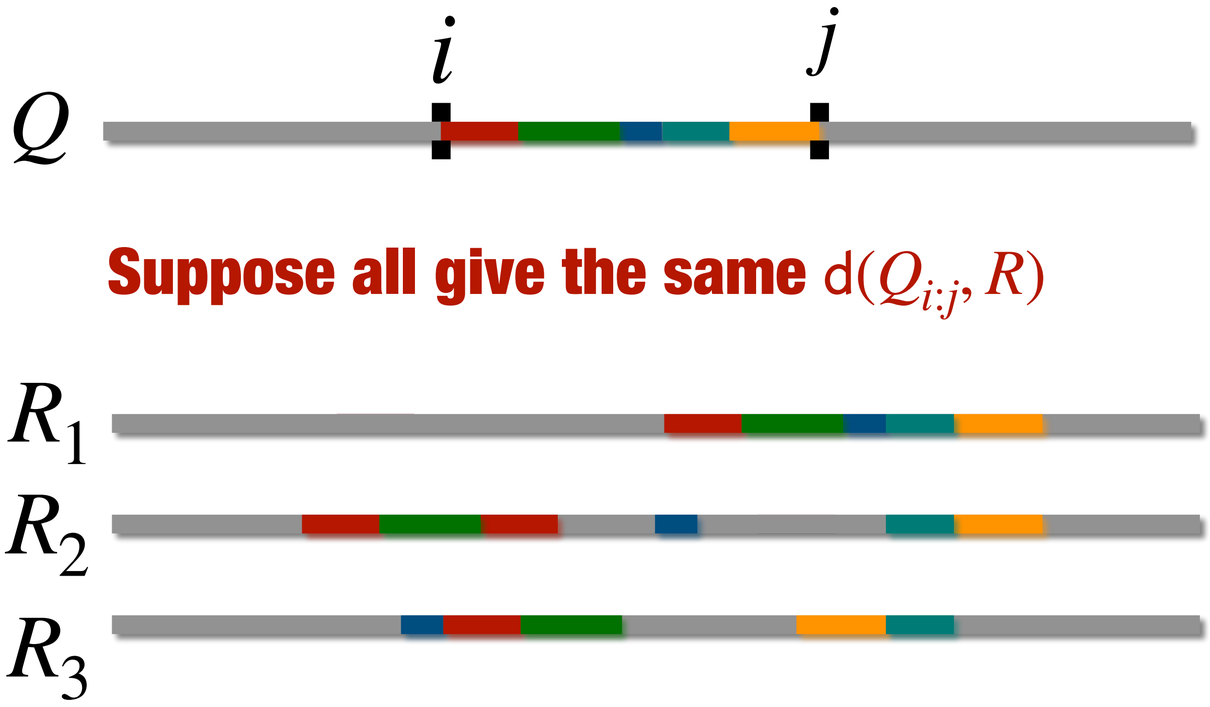
Not suitable for **highly divergent** sequences… Scalibility is a challenge…



Our focus is outlier regions

Not suitable for **highly divergent** sequences… Scalibility is a challenge…

Alternative: define a distance from **segments on Q to the entire R**!



Our focus is outlier regions

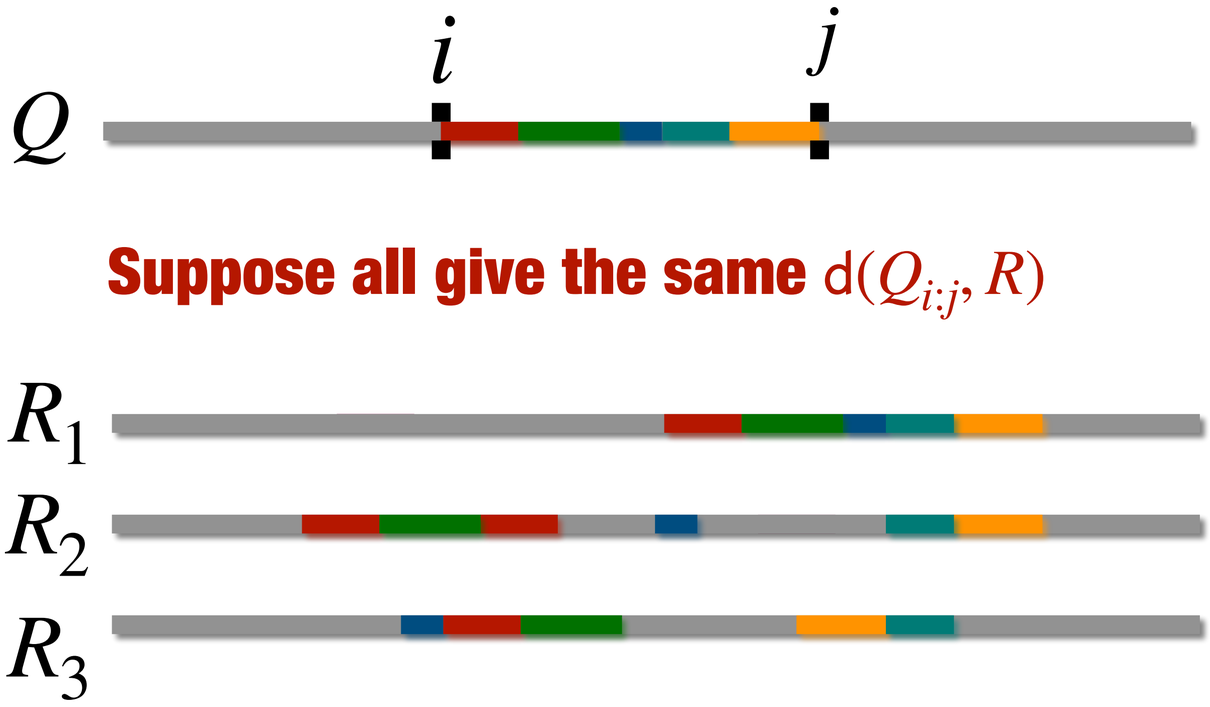
Not suitable for **highly divergent** sequences… Scalibility is a challenge…

Alternative: define a distance from **segments on Q to the entire R**!

Going even further…

Synteny-free by design

Including non-orthologous



Our focus is outlier regions

Not suitable for **highly divergent** sequences… Scalibility is a challenge…

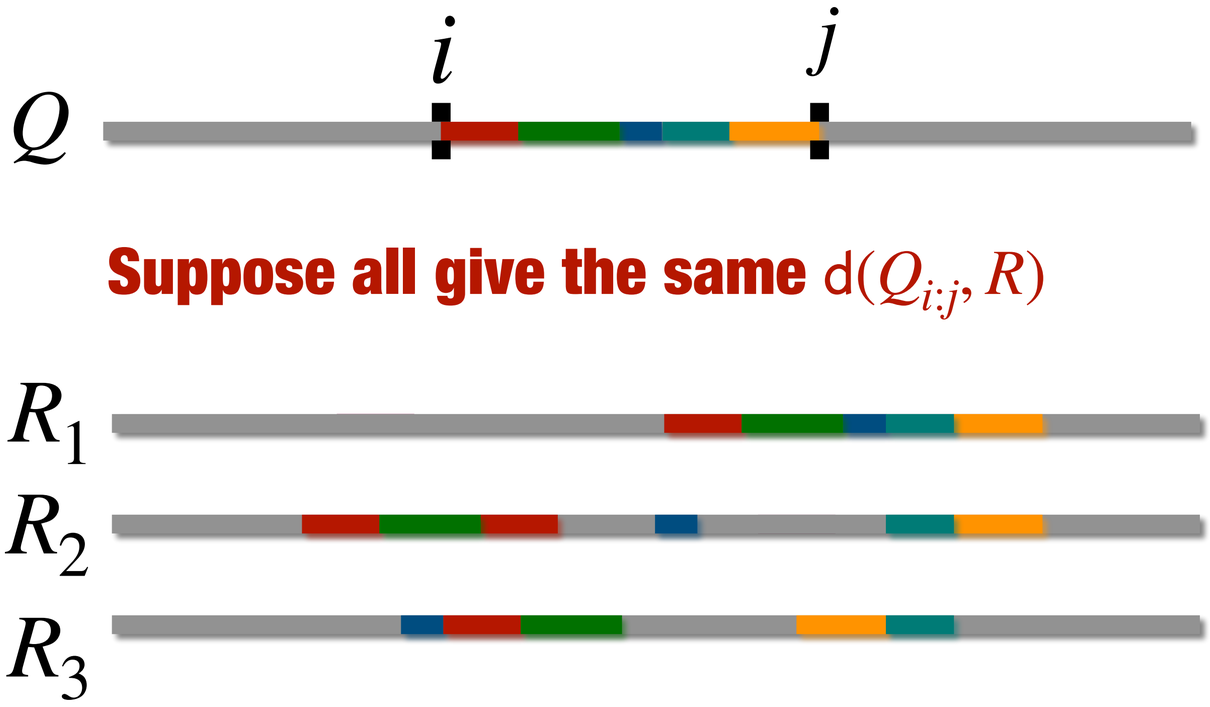
Alternative: define a distance from **segments on Q to the entire R**!

Going even further…

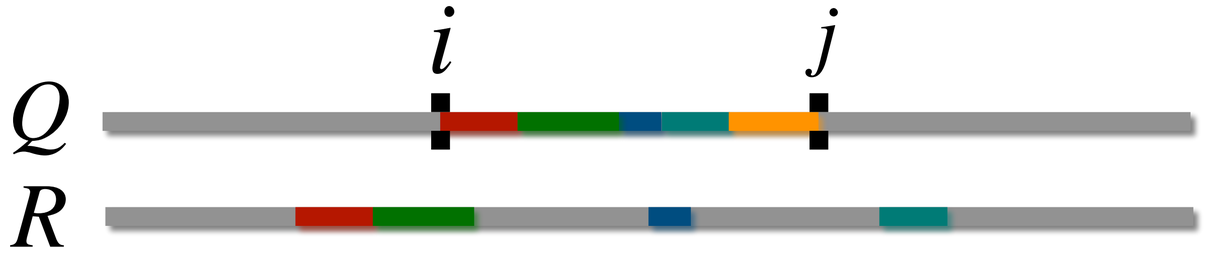
Synteny-free by design

Including non-orthologous

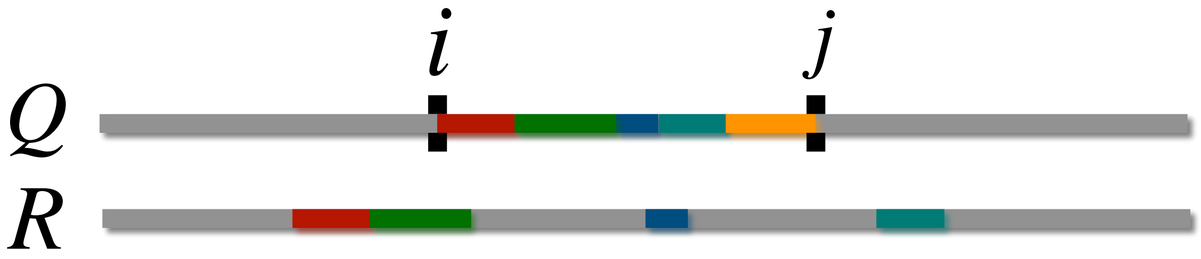
quasi-homologus→



Thinking about it recursively (for a second)

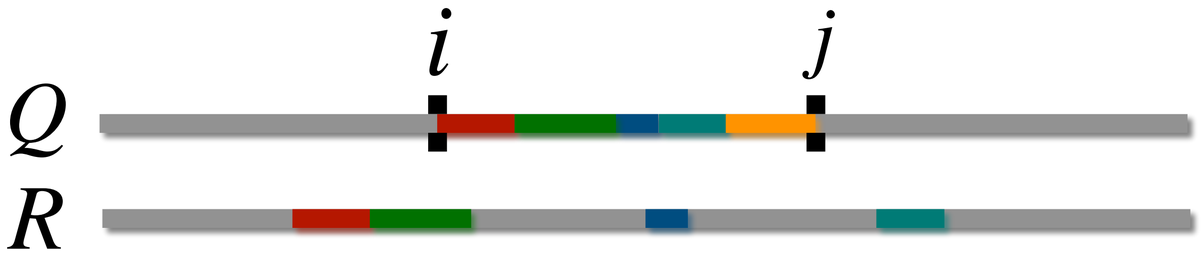


Thinking about it recursively (for a second)



$$\mathrm{d}(Q_{i:j}, R) = \sum_{q \in \{\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\}} \frac{\mathrm{length}(q)}{j - i} \mathrm{d}(q, R)$$

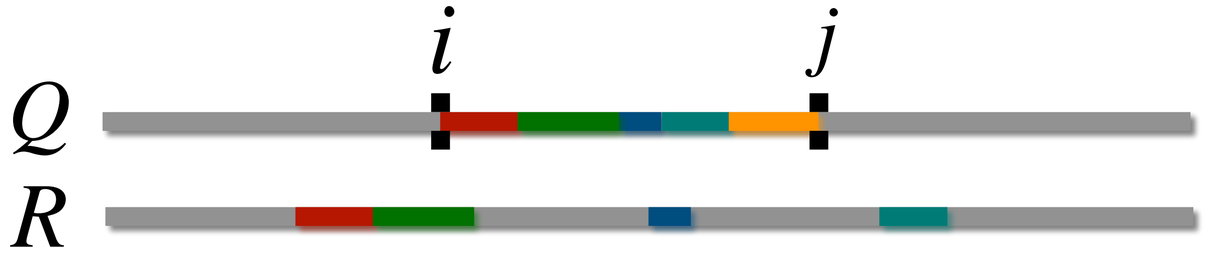
Thinking about it recursively (for a second)



$$\mathsf{d}(Q_{i:j}, R) = \sum_{q \in \{\blacksquare\blacksquare\blacksquare\blacksquare\}} \frac{\text{length}(q)}{j-i} \mathsf{d}(q, R)$$

$\mathsf{d}(\blacksquare, R) = \mathsf{d}(\blacksquare, \blacksquare)$: e.g., Hamming distance, alignment etc.

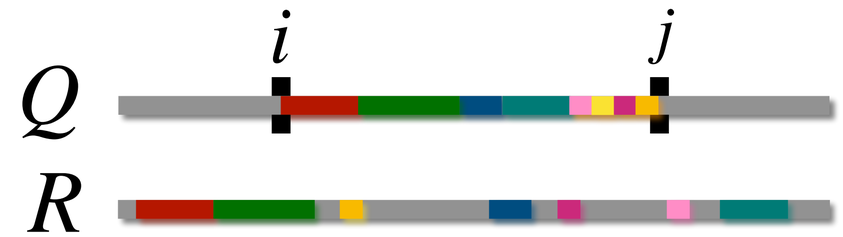
Thinking about it recursively (for a second)



$$\mathsf{d}(Q_{i:j}, R) = \sum_{q \in \{\blacksquare\}} \frac{\text{length}(q)}{j-i} \mathsf{d}(q, R)$$

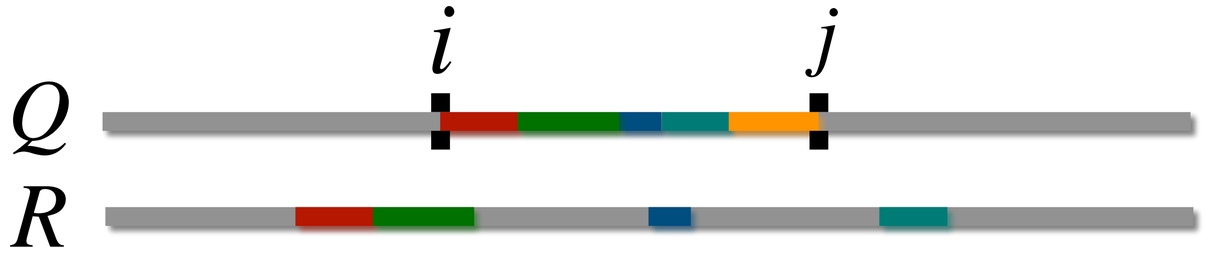
$\mathsf{d}(\underline{Q}, R) = \mathsf{d}(\underline{Q}, \underline{R})$: e.g., Hamming distance, alignment etc.

$\mathsf{d}(\underline{Q}, R)$: Recurse? →



chop into smaller pieces untill smaller segment could be mapped

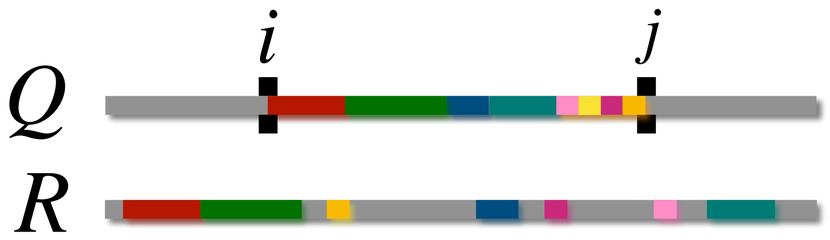
Thinking about it recursively (for a second)



$$\mathsf{d}(Q_{i:j}, R) = \sum_{q \in \{\blacksquare\}} \frac{\text{length}(q)}{j-i} \mathsf{d}(q, R)$$

$\mathsf{d}(\blacksquare, R) = \mathsf{d}(\blacksquare, \blacksquare)$: e.g., Hamming distance, alignment etc.

$\mathsf{d}(\blacksquare, R)$: Recurse? →

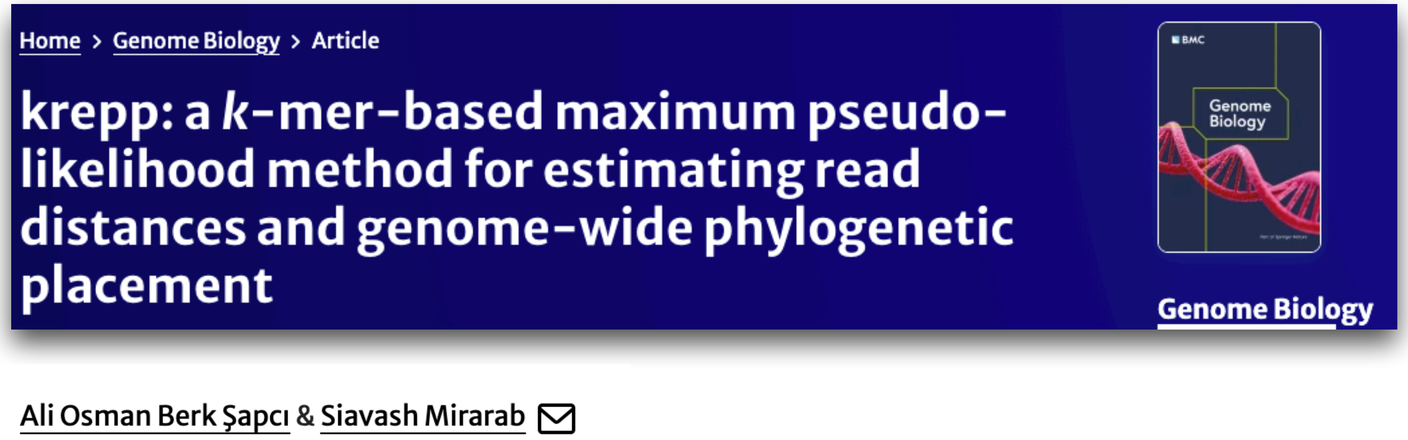


chop into smaller pieces untill smaller segment could be mapped

Approach "inductively" and we are back to k-mers (but for different reasons)…

Good news: krepp (2025)

- Searching "homologous" k-mers and measuring Hamming distances
- Estimating distances from sequences to reference genomes using k-mers



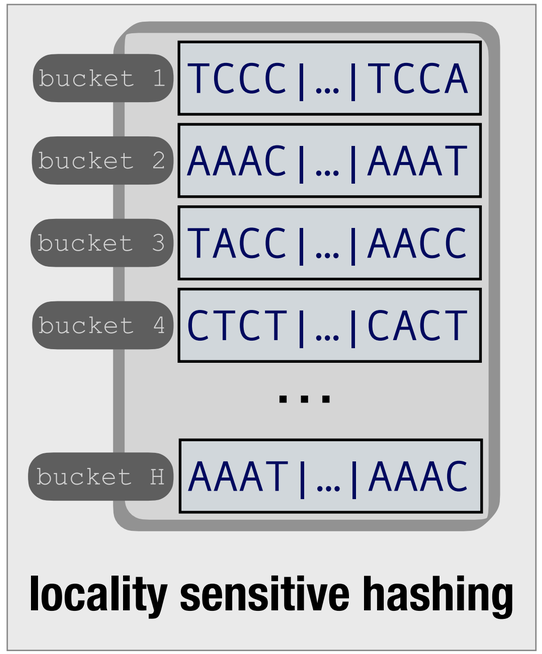
Searching for “homologous” k-mers


```
  i: ATAC
i+1:  TACC
i+2:   ACCT
i+3:    CCTA
i+4:     CTAG
                ...
j-1:                GGAC
```

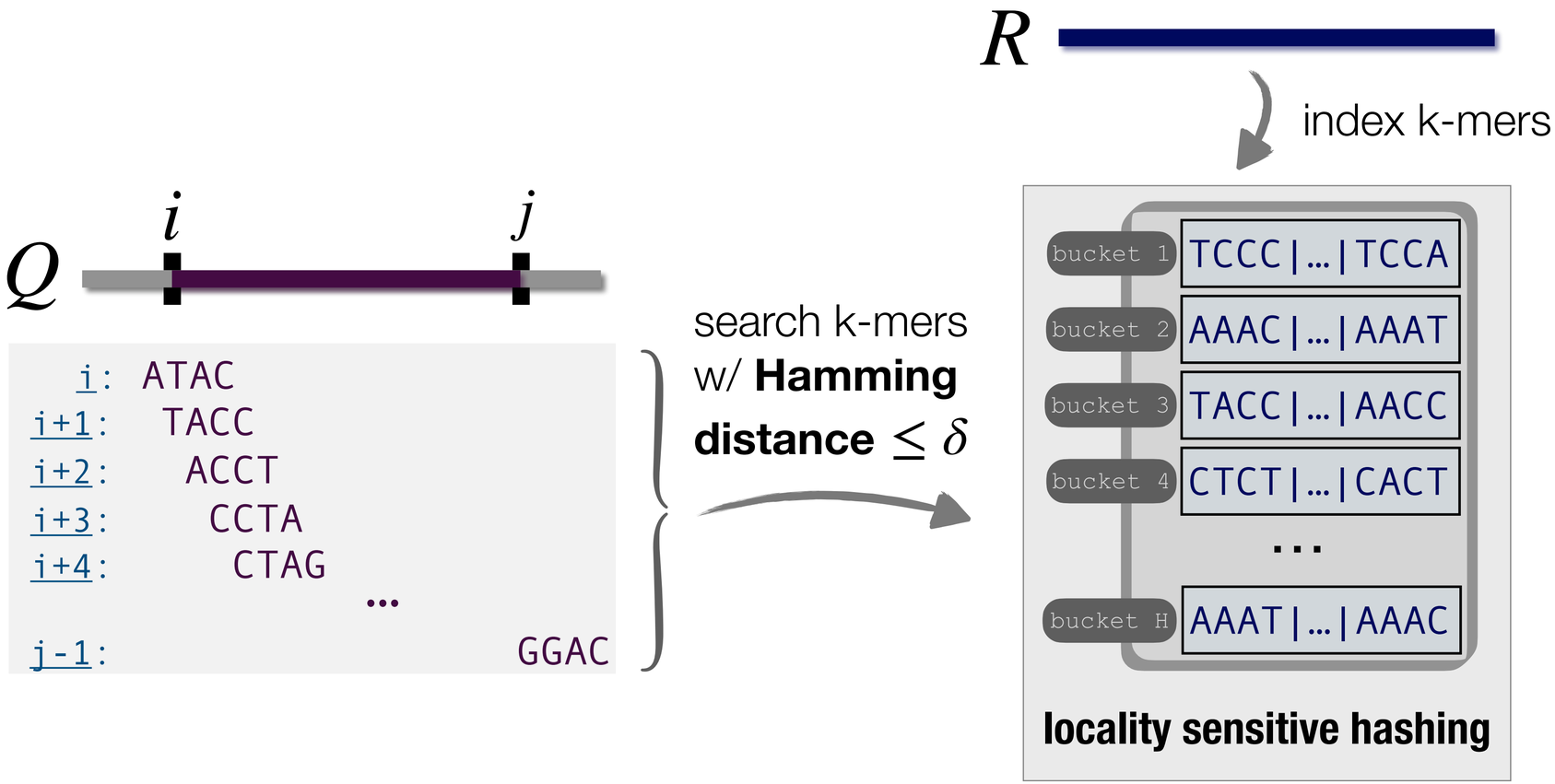
Searching for “homologous” k-mers


```
   i: ATAC
 i+1:  TACC
 i+2:   ACCT
 i+3:    CCTA
 i+4:     CTAG
                ...
 j-1:                GGAC
```

index k-mers

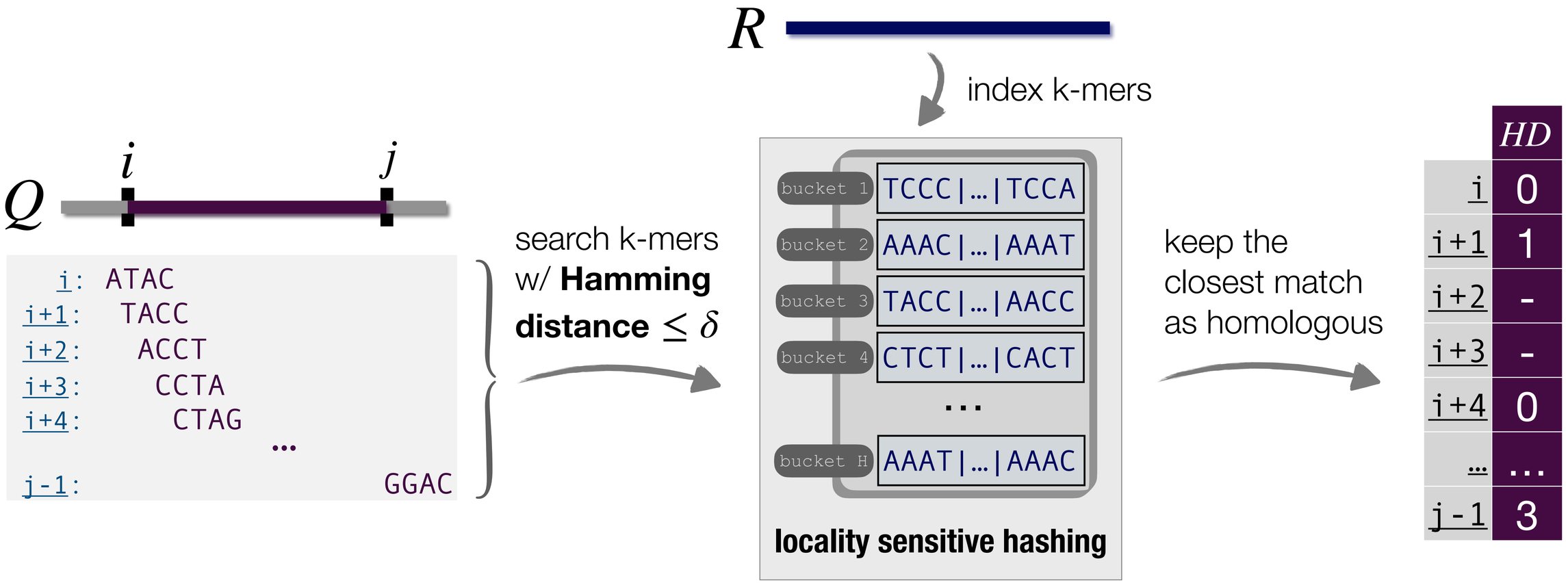


Searching for “homologous” k-mers



- threshold δ to avoid spurious matches!
- no false positives but false negatives can occur!

Searching for “homologous” k-mers



- threshold δ to avoid spurious matches!
- no false positives but false negatives can occur!

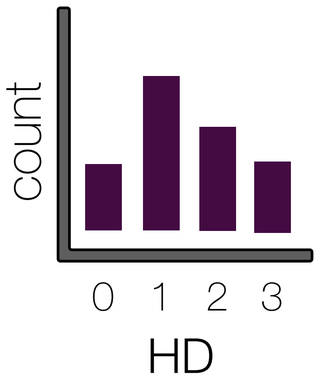
Estimating a distance from the homologous k-mers

	HD
i	0
i+1	1
i+2	-
i+3	-
i+4	0
…	…
j-1	3

Estimating a distance from the homologous k-mers

Summarize as a histogram:

	HD
i	0
i+1	1
i+2	-
i+3	-
i+4	0
…	…
j-1	3



matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j - i) - \sum_{d=0}^{\delta} v_d$

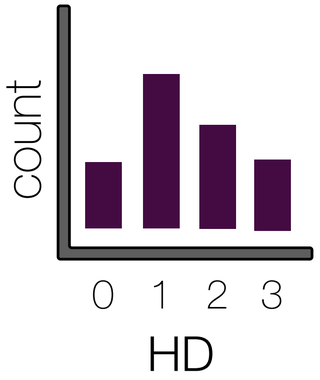
Independence assumption:

treat $Q_{i:j}$ as a bag of $j - i$ *k*-mers

Estimating a distance from the homologous k-mers

Summarize as a histogram:

	HD
i	0
i+1	1
i+2	-
i+3	-
i+4	0
…	…
j-1	3



matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j - i) - \sum_{d=0}^{\delta} v_d$

Independence assumption:

treat $Q_{i:j}$ as a bag of $j - i$ *k*-mers

Likelihood of distance D to reference R : a product over all *k*-mers

$$\mathscr{L}(D; k, h, \delta, u, \mathbf{v}) = P_{miss}(D; k, h, \delta)^u \prod_{d=0}^{\delta} P_{match}(D; d, k, h)^{v_d}$$

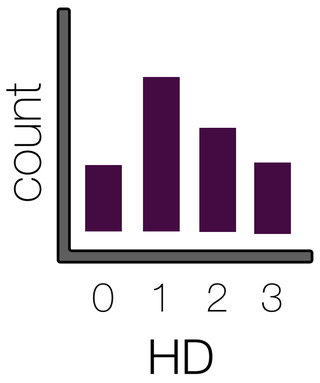
Probability of having u missses in total

Probability of having v_d matches at HD $= d$

Estimating a distance from the homologous k-mers

Summarize as a histogram:

	HD
i	0
i+1	1
i+2	-
i+3	-
i+4	0
…	…
j-1	3



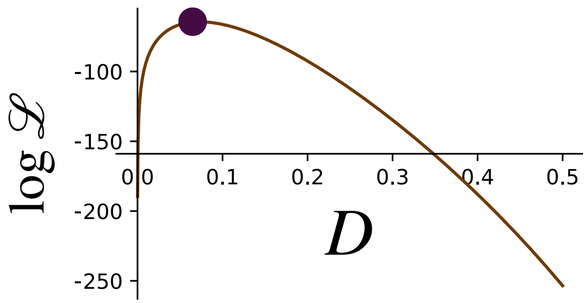
matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j - i) - \sum_{d=0}^{\delta} v_d$

Independence assumption:

treat $Q_{i:j}$ as a bag of $j - i$ *k*-mers

Compute the likelihood of $Q_{i:j}$ having distance D to R



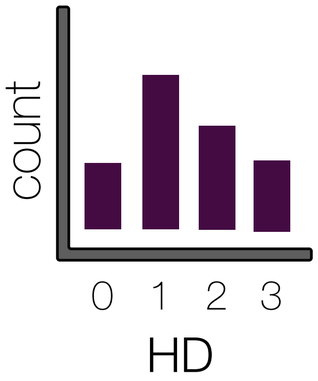
Likelihood of distance D to reference R : a product over all *k*-mers

$$\mathscr{L}(D; k, h, \delta, u, \mathbf{v}) = P_{miss}(D; k, h, \delta)^u \prod_{d=0}^{\delta} P_{match}(D; d, k, h)^{v_d}$$

Probability of having u missses in total

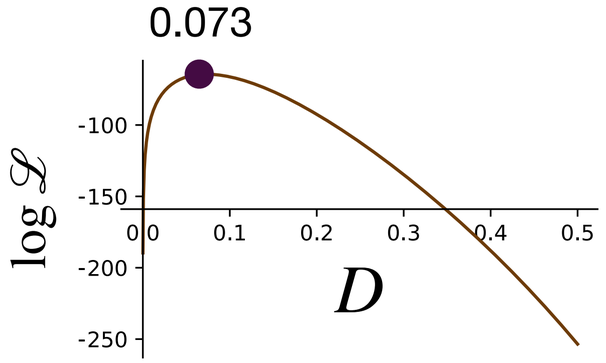
Probability of having v_d matches at HD $= d$

Maximum likelihood estimation of distances



Optimize $-\log \mathscr{L}$ w.r.t. D for each reference R:

$$\arg\max_D \; u \log P_{miss}(D; k, h, \delta) \sum_{d=0}^{\delta} v_d \log P_{match}(D; d, k, h)$$



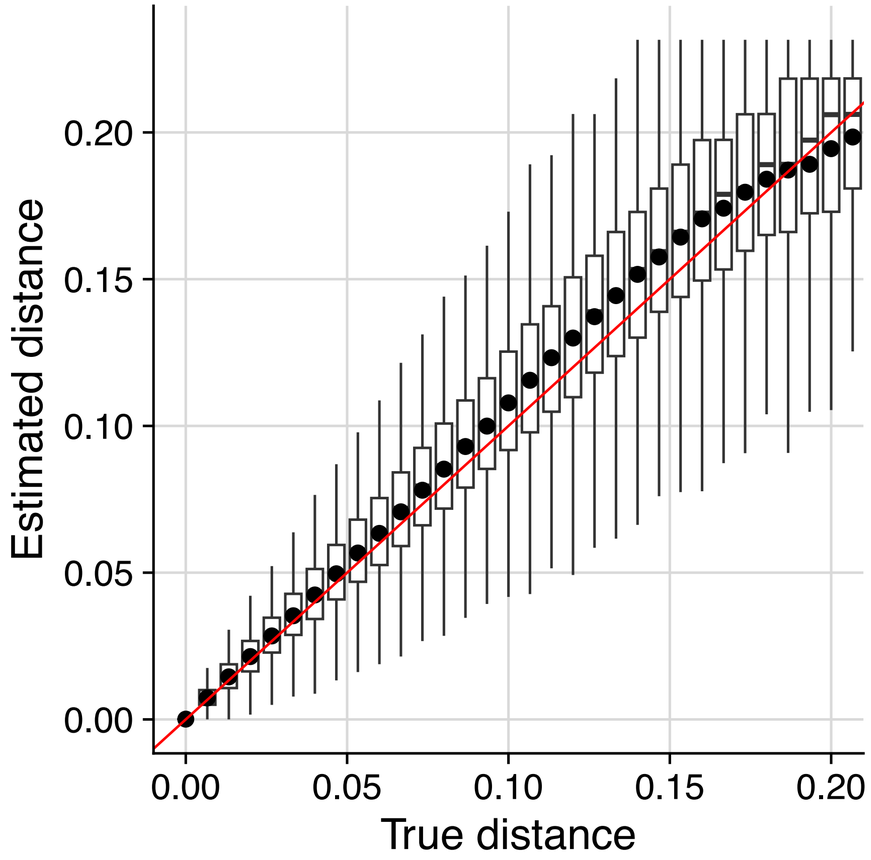
convex w/ sensible parameters

Maximum pseudo-likelihood distances are accurate!

29-mer minimizers of 35-mers

- Simulation experiments (true read distances)
- Highly accurate (despite some noise)
- Slight overestimation bias for high distances

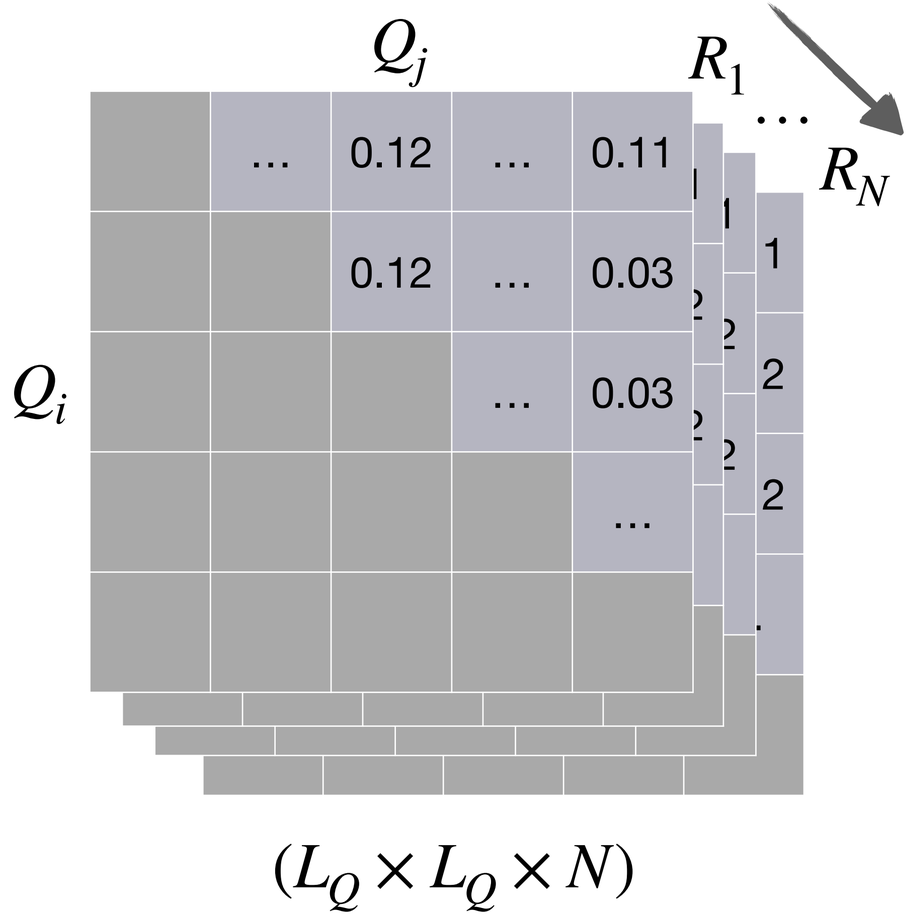
(Hamming distance) / (seq. length)



Can a distance for every interval be computed?

From $\mathcal{O}(L_R^2 L_Q^2)$ distances to $\mathcal{O}(L_Q^2)$ per reference!

Query segment to reference genome:

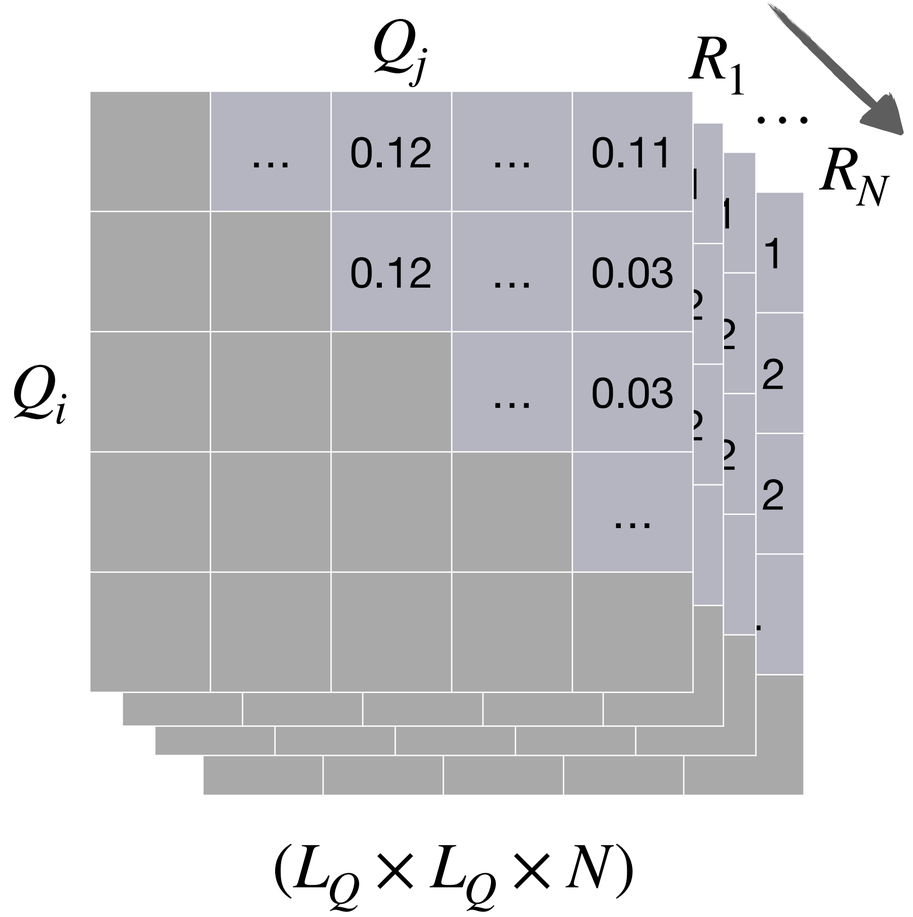


Can a distance for every interval be computed?

From $\mathcal{O}(L_R^2 L_Q^2)$ distances to $\mathcal{O}(L_Q^2)$ per reference!

For a large $\mathscr{R}$, $\mathcal{O}(L_Q^2 |\mathscr{R}|)$ distances is **still not feasible**

Query segment to reference genome:



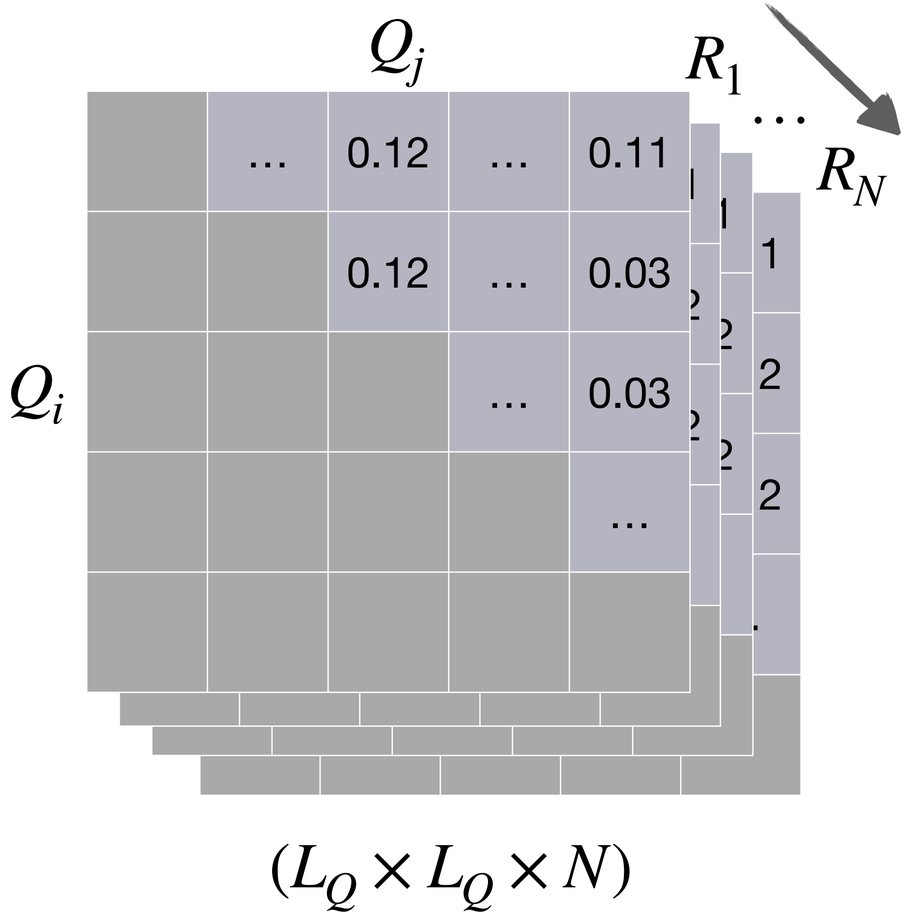
Can a distance for every interval be computed?

From $\mathcal{O}(L_R^2 L_Q^2)$ distances to $\mathcal{O}(L_Q^2)$ per reference!

For a large $\mathscr{R}$, $\mathcal{O}(L_Q^2 |\mathscr{R}|)$ distances is **still not feasible**

Simpler problems: given Q & R, thresholds Δ and τ

Query segment to reference genome:



$(L_Q \times L_Q \times N)$

Can a distance for every interval be computed?

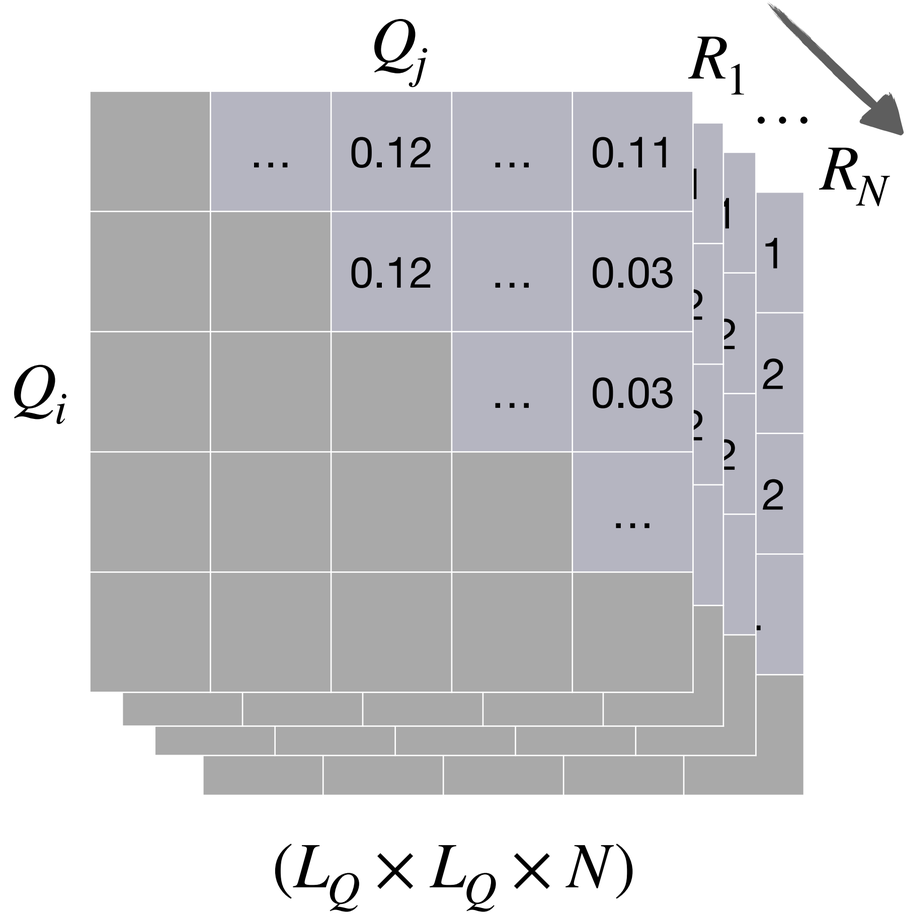
From $\mathcal{O}(L_R^2 L_Q^2)$ distances to $\mathcal{O}(L_Q^2)$ per reference!

For a large $\mathscr{R}$, $\mathcal{O}(L_Q^2 |\mathscr{R}|)$ distances is **still not feasible**

Simpler problems: given Q & R, thresholds Δ and τ

Decide whether an interval (i, j) satisfies $\mathsf{d}(Q_{i:j}, R) < \Delta$.

Query segment to reference genome:



Can a distance for every interval be computed?

From $\mathcal{O}(L_R^2 L_Q^2)$ distances to $\mathcal{O}(L_Q^2)$ per reference!

For a large $\mathscr{R}$, $\mathcal{O}(L_Q^2 |\mathscr{R}|)$ distances is **still not feasible**

Simpler problems: given Q & R, thresholds Δ and τ

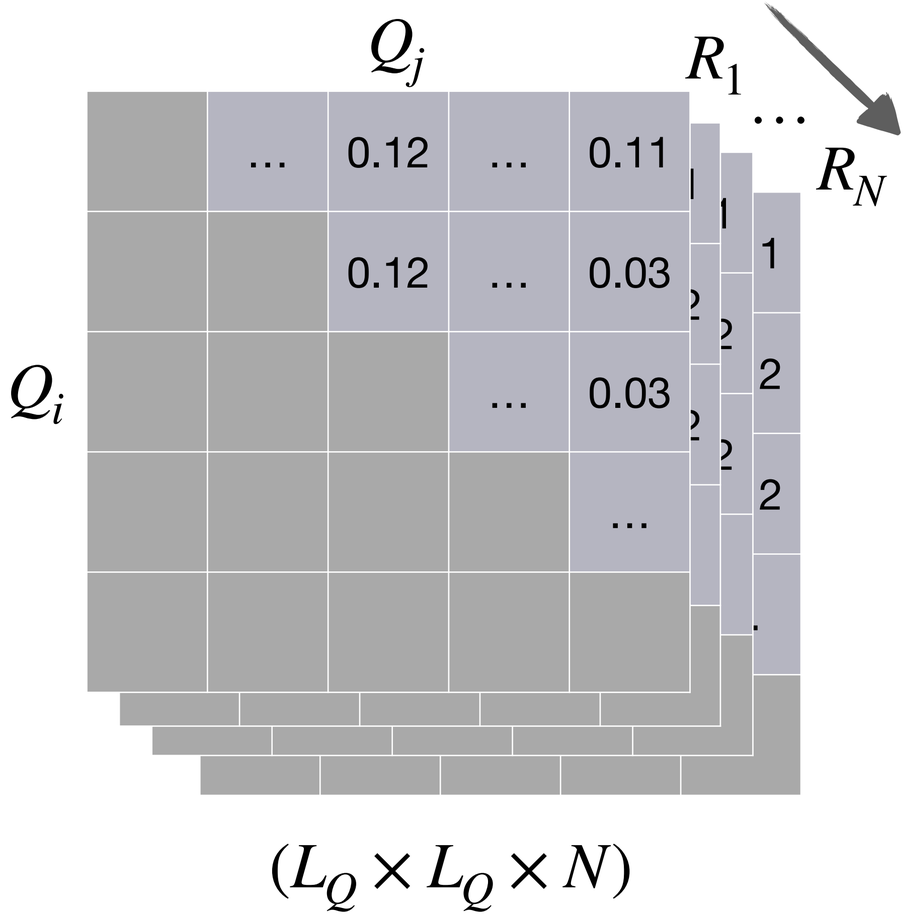
Decide whether an interval (i, j) satisfies $\mathsf{d}(Q_{i:j}, R) < \Delta$.

Enumerate all **maximal intervals** (i, j) such that

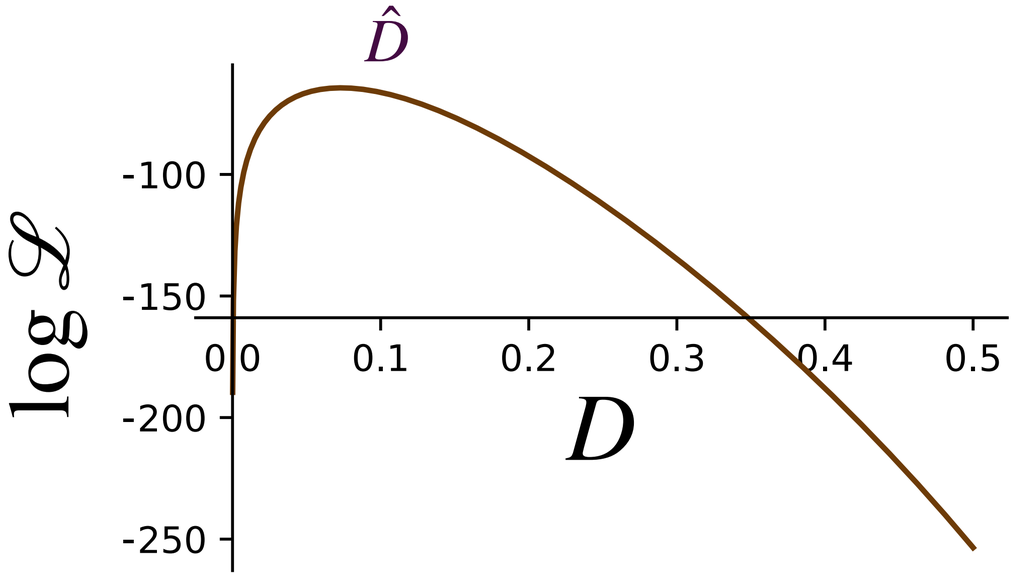
$$\mathsf{d}(Q_{i:j}, R) < \Delta \text{ and } j - i \geq \tau,$$

$$\mathsf{d}(Q_{a:b}, R) \geq \delta \text{ for } a \leq i \leq j \leq b, (i, j) \neq (a, b).$$

Query segment to reference genome:

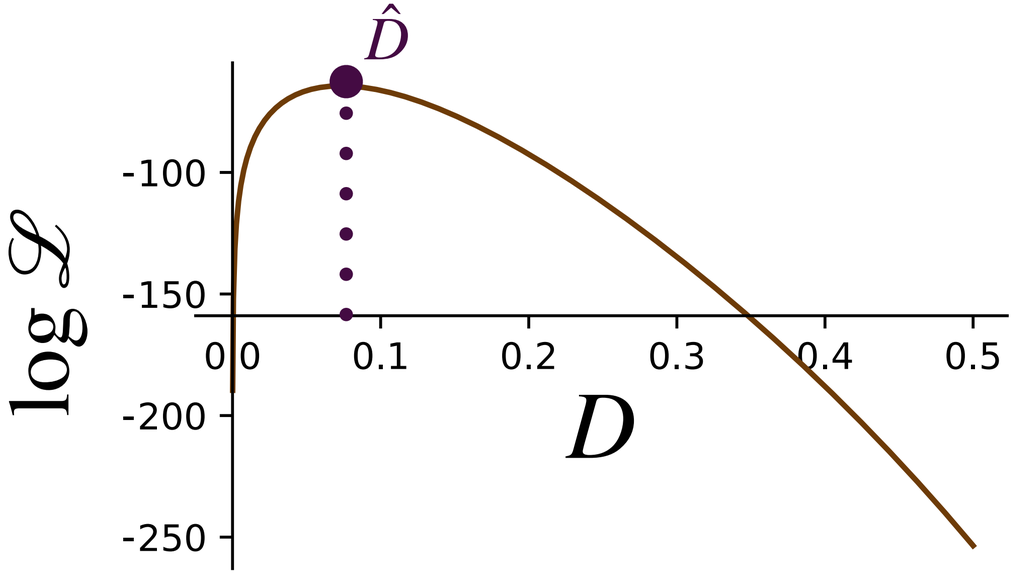


gdiff: distance-based pattern detection using the derivatives



gdiff: distance-based pattern detection using the derivatives

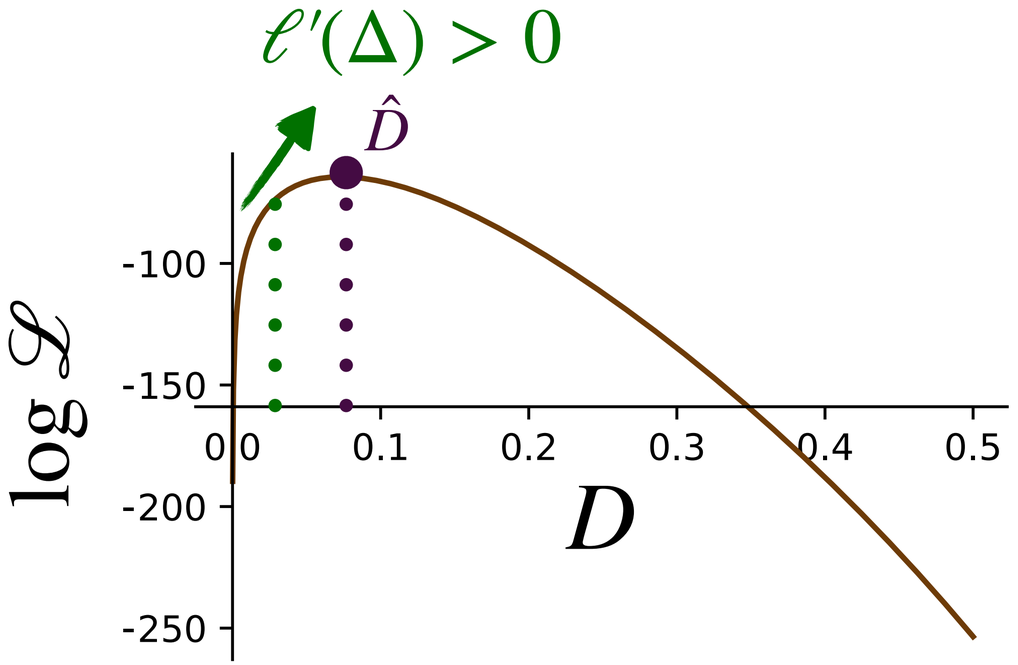
MLE distance $\hat{D} = \Delta \rightarrow \ell'(\Delta) = 0$



gdiff: distance-based pattern detection using the derivatives

MLE distance $\hat{D} = \Delta \rightarrow \ell'(\Delta) = 0$

If $\Delta < \hat{D} \rightarrow \ell'(\Delta) > 0$

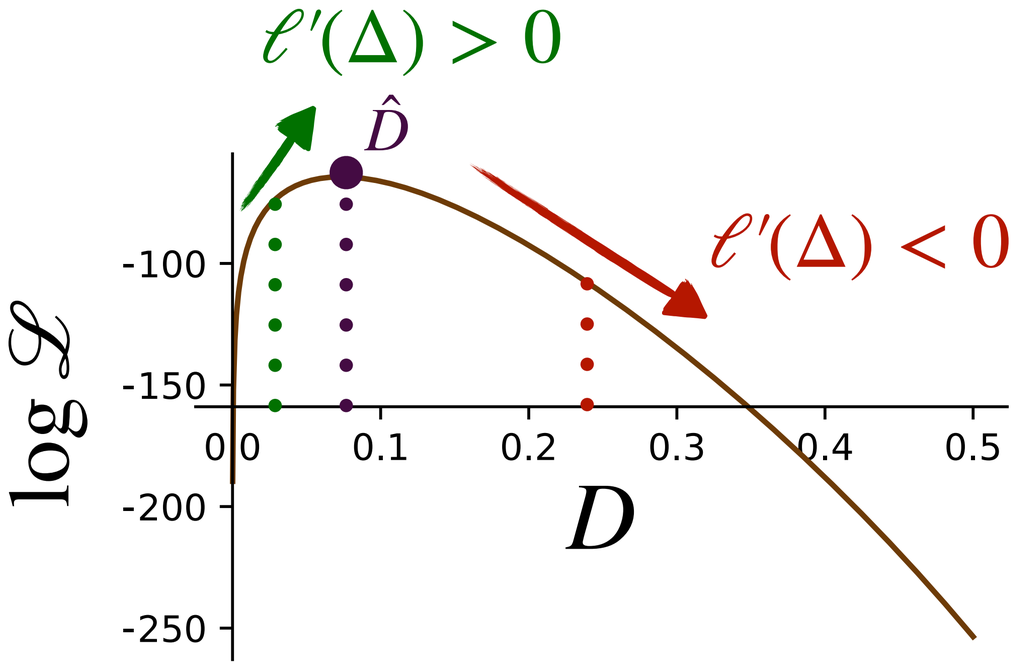


gdiff: distance-based pattern detection using the derivatives

MLE distance $\hat{D} = \Delta \rightarrow \ell'(\Delta) = 0$

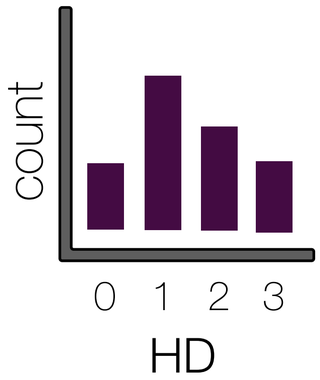
If $\Delta < \hat{D} \rightarrow \ell'(\Delta) > 0$

If $\Delta > \hat{D} \rightarrow \ell'(\Delta) < 0$



No optimization: just a linear combination over k-mers

Compute the derivative
at the threshold $D = \Delta$.



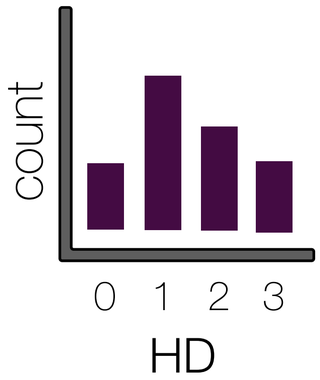
matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j - i) - \sum_{d=0}^{\delta} v_d$

No optimization: just a linear combination over k-mers

Compute the derivative at the threshold $D = \Delta$.

$$\ell'(D) = u\frac{\rho\left(\sum_{d=0}^{k} D^d(1-D)^{k-d}\binom{k}{d}\left(\frac{d-kD}{D(1-D)}\right)P_\delta(d)\right)}{1-\rho+\rho\left(\sum_{d=0}^{k} D^d(1-D)^{k-d}\binom{k}{d}P_\delta(d)\right)} + \sum_{d=0}^{\delta} v_d\left(\frac{d-kD}{D(1-D)}\right)$$



matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j-i) - \sum_{d=0}^{\delta} v_d$

No optimization: just a linear combination over k-mers

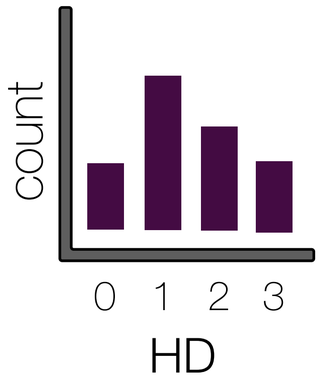
Simply a linear combination over the histogram!

Compute the derivative at the threshold $D = \Delta$.

$$\ell'(D) = u \frac{\rho\left(\sum_{d=0}^{k} D^d (1-D)^{k-d} \binom{k}{d} \left(\frac{d-kD}{D(1-D)}\right) P_\delta(d)\right)}{1-\rho+\rho\left(\sum_{d=0}^{k} D^d (1-D)^{k-d} \binom{k}{d} P_\delta(d)\right)} + \sum_{d=0}^{\delta} v_d \left(\frac{d-kD}{D(1-D)}\right)$$

contribution to the derivate per missed k-mer

per matched k-mer



matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j-i) - \sum_{d=0}^{\delta} v_d$

No optimization: just a linear combination over k-mers

Simply a linear combination over the histogram!

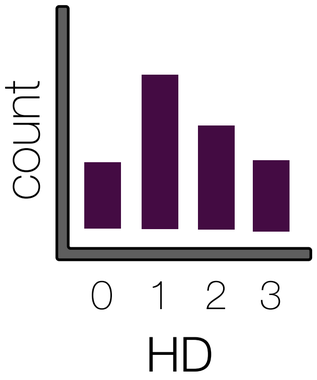
Compute the derivative at the threshold $D = \Delta$.

precomputed: $c(-, \Delta)$ $c(d; \Delta)$

$$\ell'(D) = u \frac{\rho\left(\sum_{d=0}^{k} D^d (1-D)^{k-d} \binom{k}{d} \left(\frac{d-kD}{D(1-D)}\right) P_\delta(d)\right)}{1-\rho+\rho\left(\sum_{d=0}^{k} D^d (1-D)^{k-d} \binom{k}{d} P_\delta(d)\right)} + \sum_{d=0}^{\delta} v_d \left(\frac{d-k\Delta}{\Delta(1-\Delta)}\right)$$

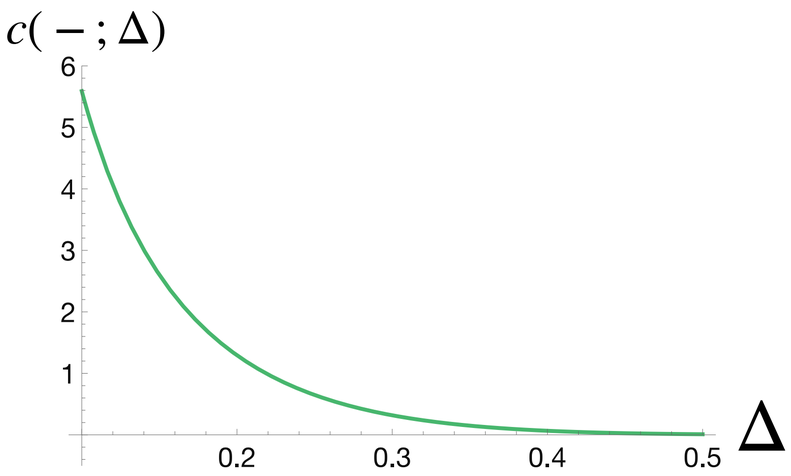
contribution to the derivate per missed k-mer

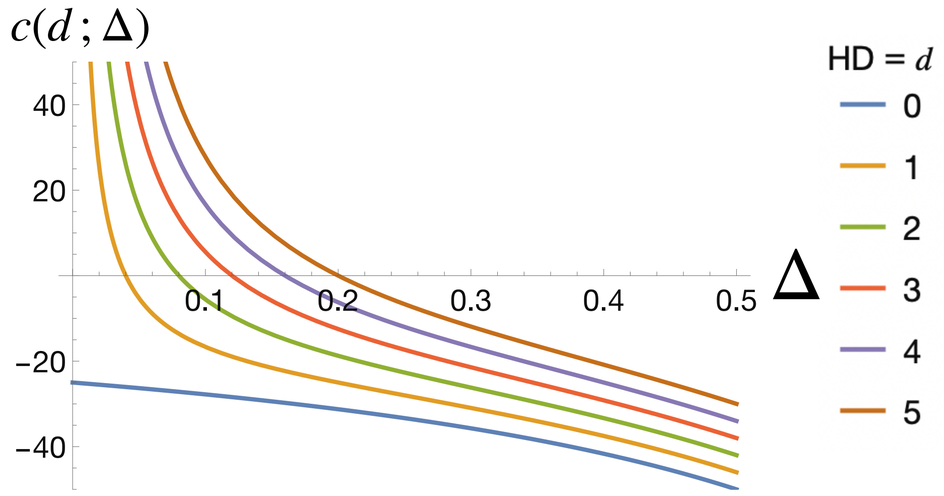
per matched k-mer



matches: $\mathbf{v} = [v_0, v_1, \ldots, v_\delta]$

misses: $u = (j-i) - \sum_{d=0}^{\delta} v_d$





Each *k*-mer adds a constant to the derivate.

Solving the decision problem after a single pass


```
   i: ATAC
 i+1:  TACC
 i+2:   ACCT
 i+3:    CCTA
 i+4:     CTAG
                ...
 j-1:               GGAC
```

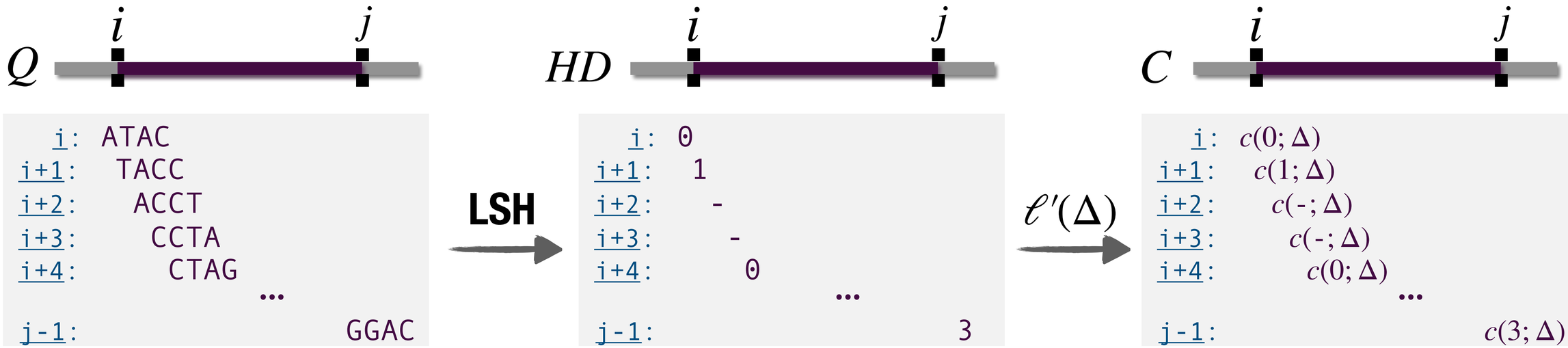
Solving the decision problem after a single pass


```
  i: ATAC
i+1:  TACC
i+2:   ACCT
i+3:    CCTA
i+4:     CTAG
              ...
j-1:              GGAC
```

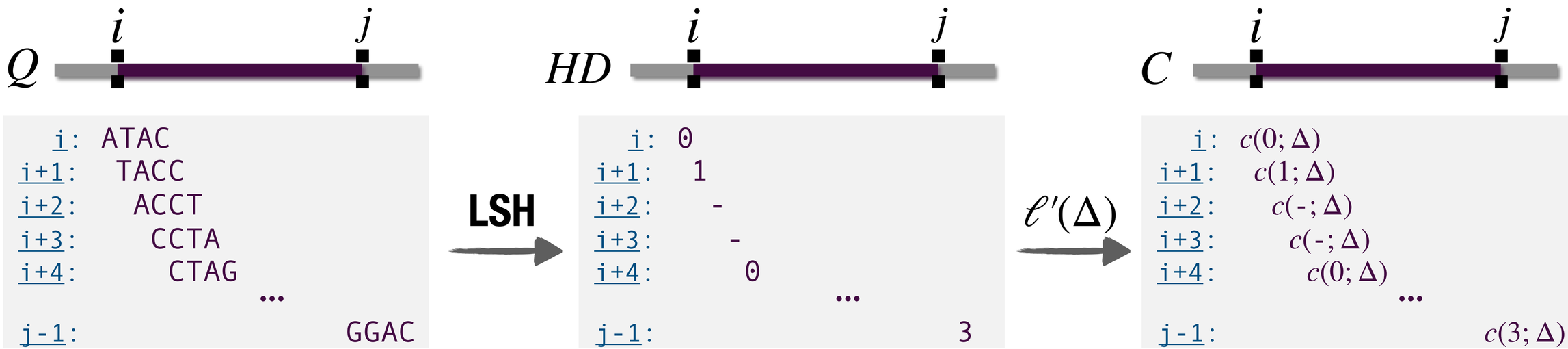
LSH

```
  i: 0
i+1:  1
i+2:   -
i+3:    -
i+4:     0
              ...
j-1:              3
```

Solving the decision problem after a single pass

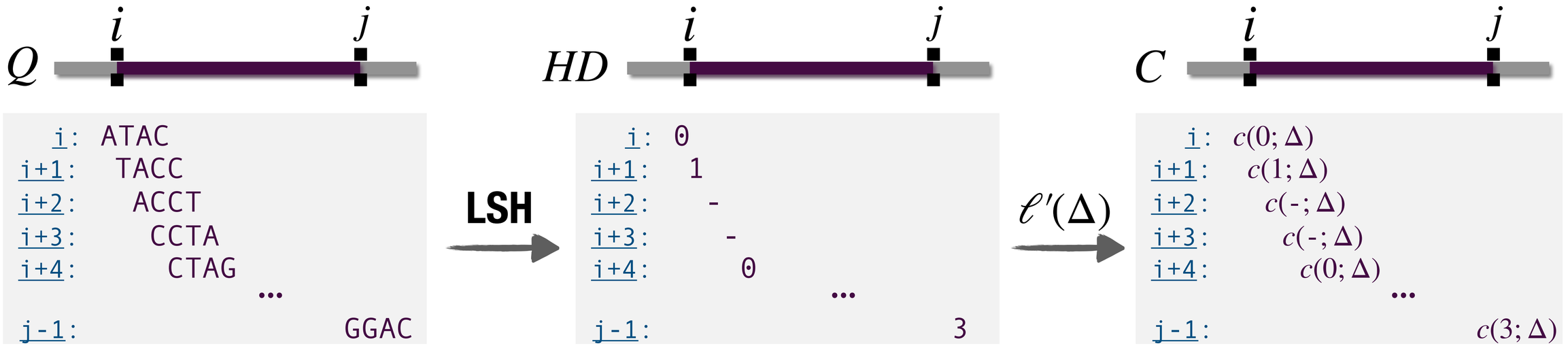


Solving the decision problem after a single pass



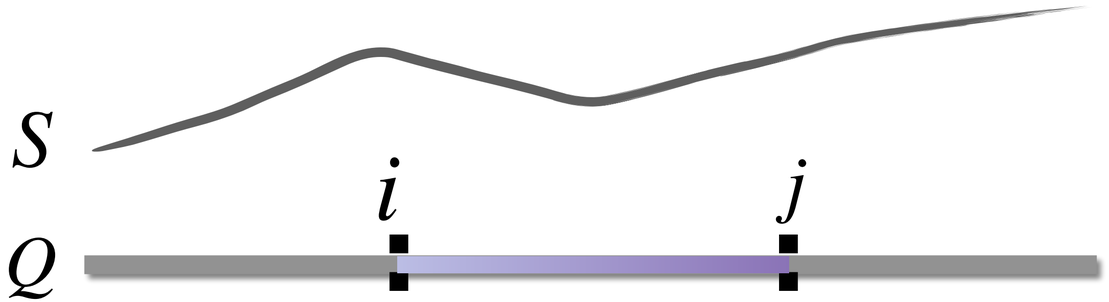
x	HD	$C(x)$
i	0	$c(0;\Delta)$
i+1	1	$c(1;\Delta)$
i+2	-	$c(-;\Delta)$
i+3	-	$c(-;\Delta)$
i+4	0	$c(0;\Delta)$
...	...	...
j-1	3	$c(3;\Delta)$

Solving the decision problem after a single pass

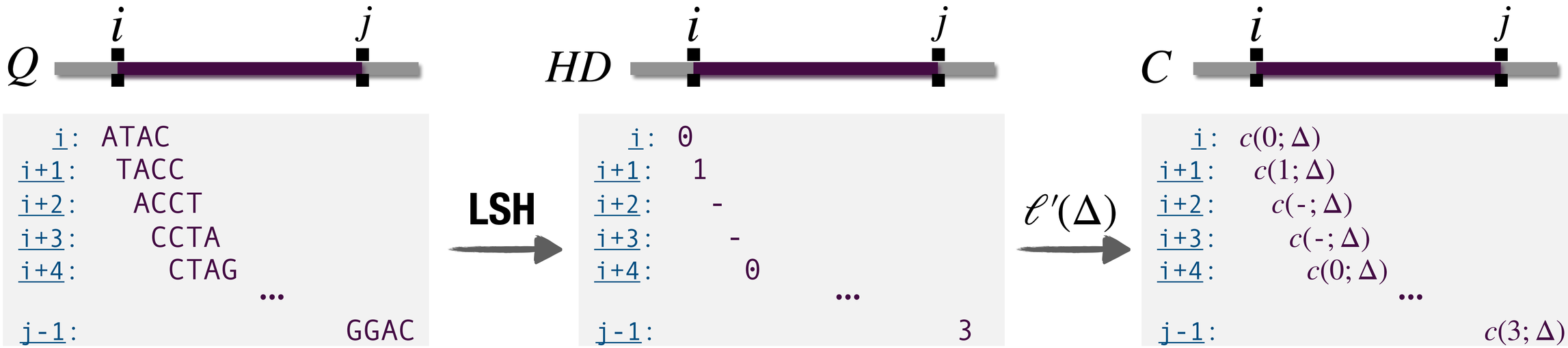


x	HD	$C(x)$
i	0	$c(0;\Delta)$
i+1	1	$c(1;\Delta)$
i+2	-	$c(-;\Delta)$
i+3	-	$c(-;\Delta)$
i+4	0	$c(0;\Delta)$
...	...	...
j-1	3	$c(3;\Delta)$

Compute the prefix-sum array $S = (s_1, \ldots, s_{N+1})$ where $s_i = \sum_{i'=1}^{i} C(x_{i'})$.

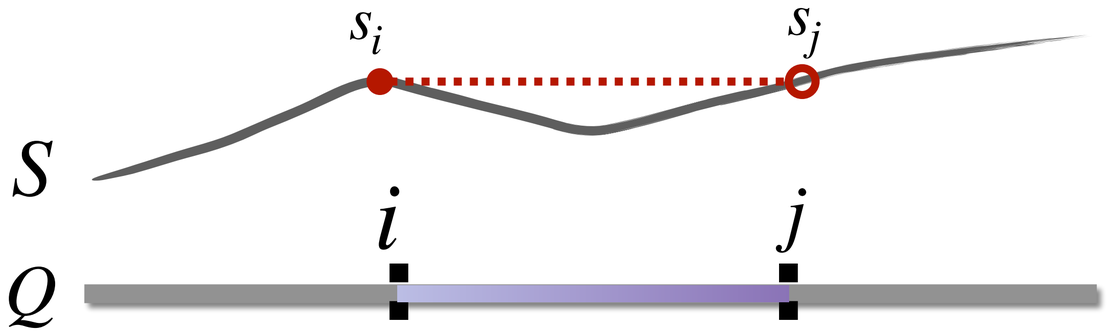


Solving the decision problem after a single pass



x	HD	$C(x)$
i	0	$c(0;\Delta)$
i+1	1	$c(1;\Delta)$
i+2	-	$c(-;\Delta)$
i+3	-	$c(-;\Delta)$
i+4	0	$c(0;\Delta)$
...	...	...
j-1	3	$c(3;\Delta)$

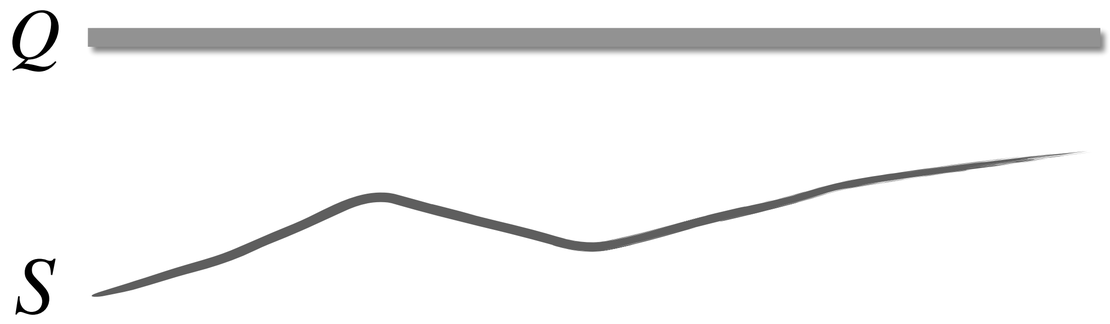
Compute the prefix-sum array $S = (s_1, \ldots, s_{N+1})$ where $s_i = \sum_{i'=1}^{i} C(x_{i'})$.



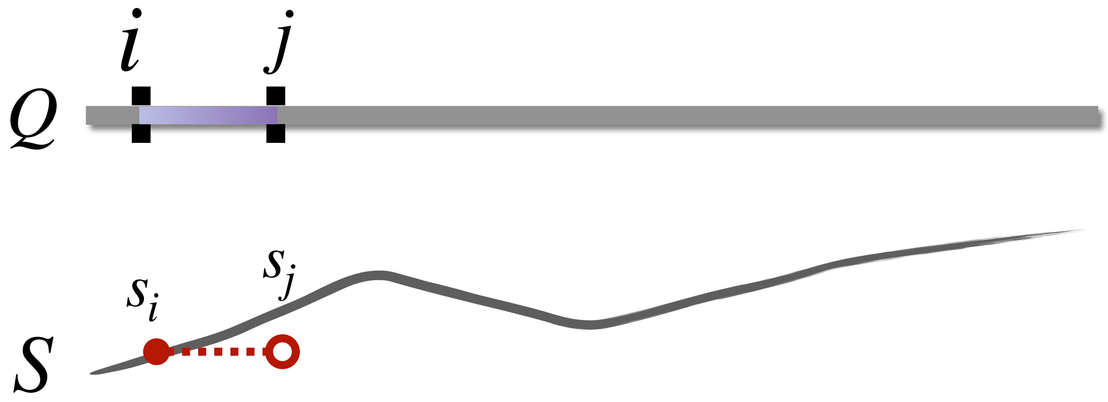
For any given interval $[i, j)$;
$\mathrm{d}(Q_{i:j}, R) < \Delta$ if $s_i > s_j$!

(constant time)

Constant time distance queries

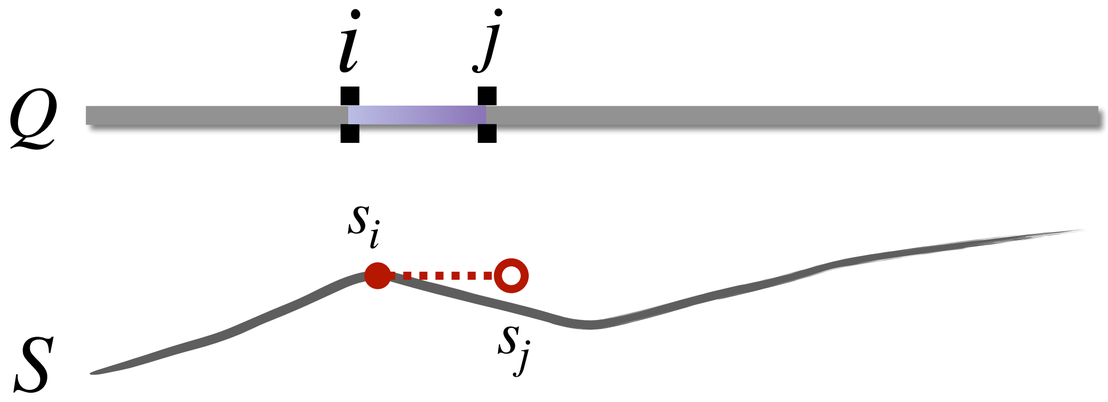


Constant time distance queries



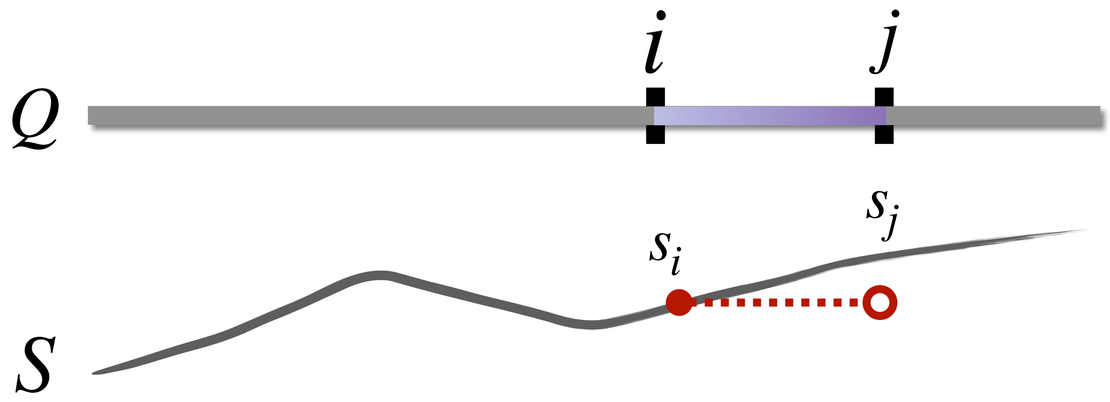
$$s_i < s_j \rightarrow \mathsf{d}(Q_{i:j}, R) > \Delta$$

Constant time distance queries



$$s_i > s_j \rightarrow \mathrm{d}(Q_{i:j}, R) < \Delta$$

Constant time distance queries



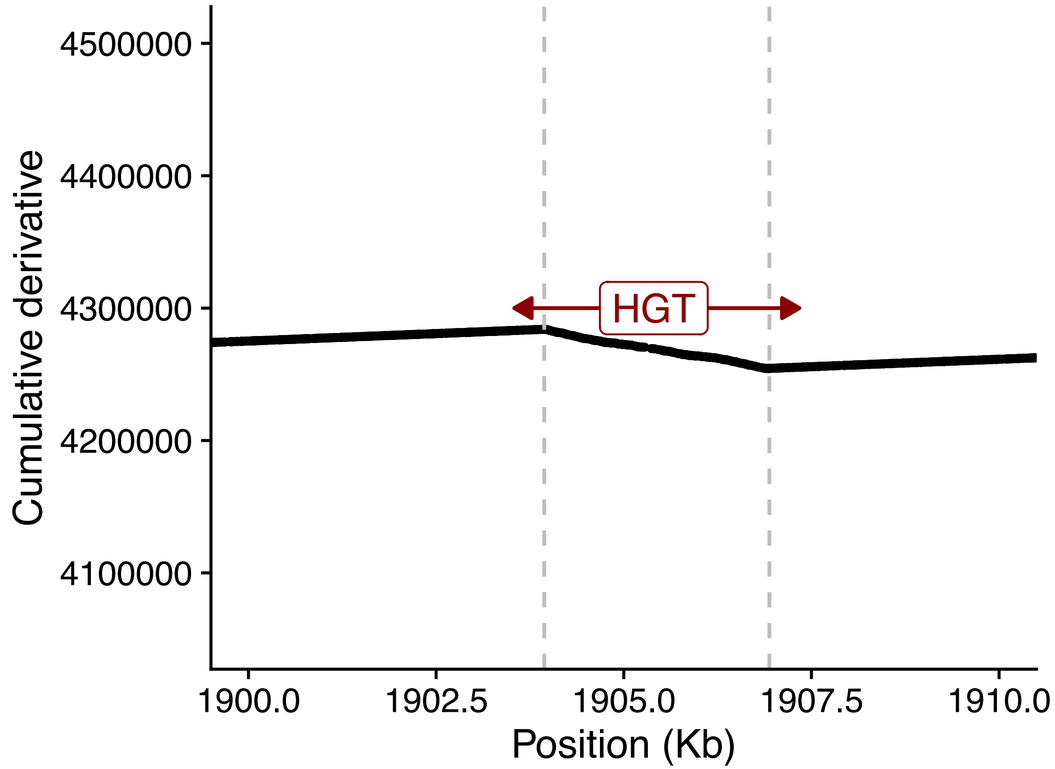
$$s_i < s_j \rightarrow \mathsf{d}(Q_{i:j}, R) > \Delta$$

An example of horizontal gene transfer

An “archaic” HGT event simulated using Zombi.

Genome-wide distance **32%**, gene distance **6%**

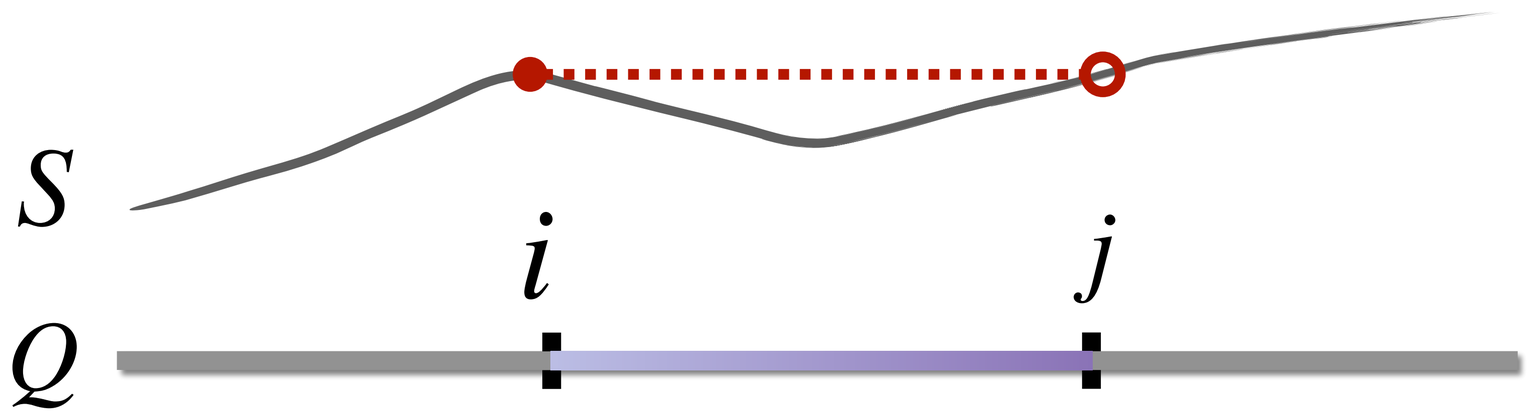
$$\Delta = 10\,\%$$



What about enumerating all the intervals?

Find all maximal intervals (i, j) with $\mathsf{d}(Q_{i:j}, R) < \Delta$ (wlog) and $j - i \geq \tau$:

Not contained in any other larger interval

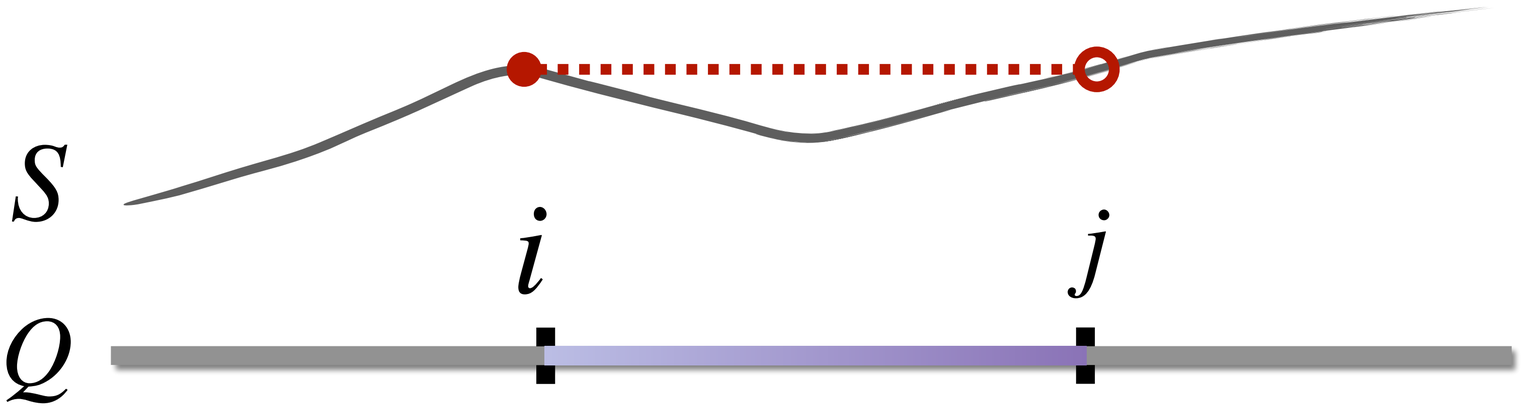


What about enumerating all the intervals?

Find all maximal intervals (i, j) with $\mathsf{d}(Q_{i:j}, R) < \Delta$ (wlog) and $j - i \geq \tau$:

Not contained in any other larger interval

1. Left maximal if s_i is a prefix maxima of S
2. Right maximal if s_j is a suffix minima of S

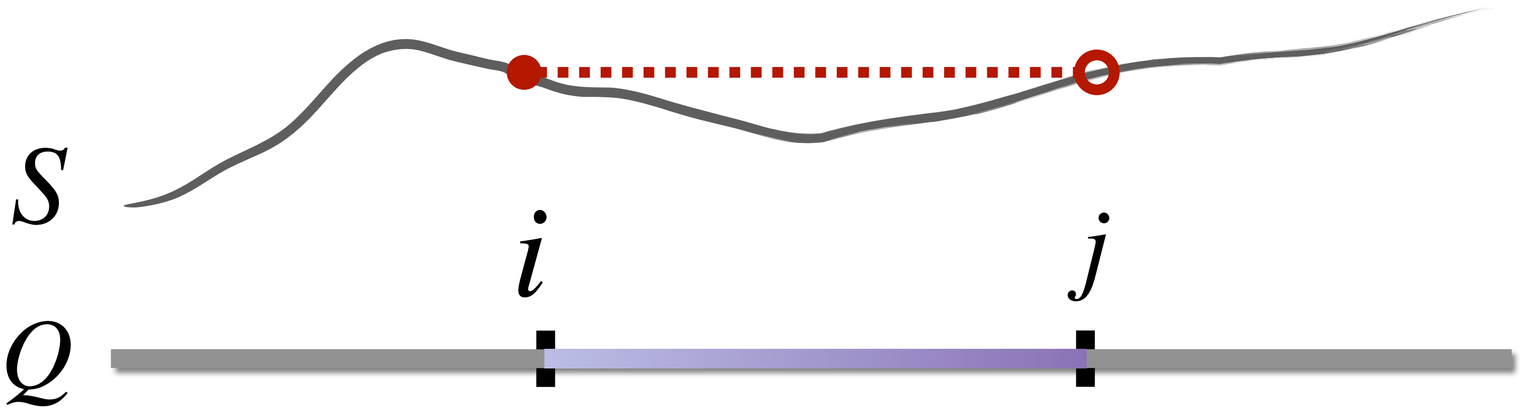


What about enumerating all the intervals?

Find all maximal intervals (i, j) with $\mathsf{d}(Q_{i:j}, R) < \Delta$ (wlog) and $j - i \geq \tau$:

Not contained in any other larger interval

1. Left maximal if s_i is a prefix maxima of S
2. Right maximal if s_j is a suffix minima of S

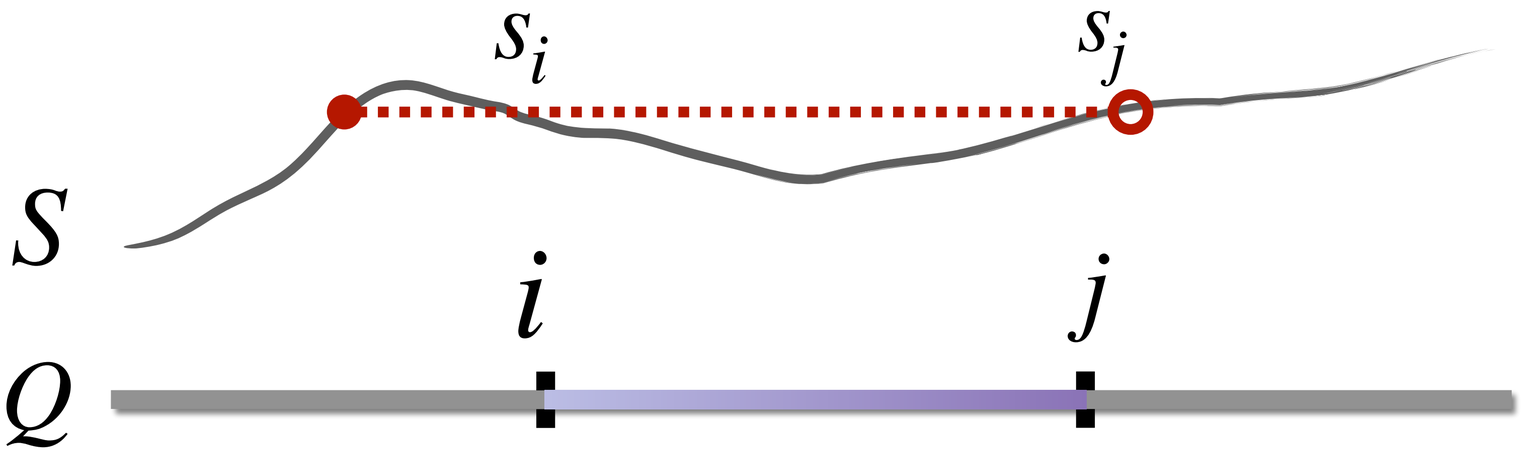


What about enumerating all the intervals?

Find all maximal intervals (i, j) with $\mathsf{d}(Q_{i:j}, R) < \Delta$ (wlog) and $j - i \geq \tau$:

Not contained in any other larger interval

1. Left maximal if s_i is a prefix maxima of S
2. Right maximal if s_j is a suffix minima of S



We can extend $[i, j)$!

A linear time algorithm

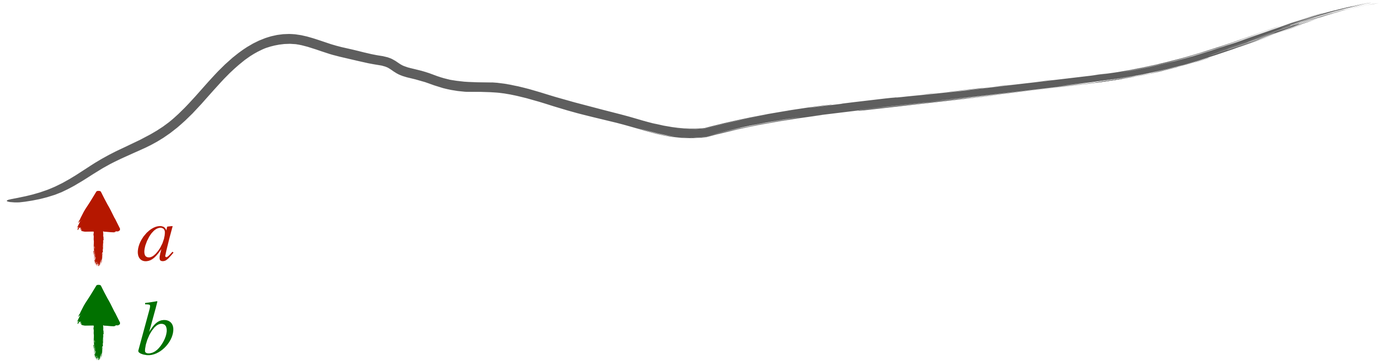
Two pointers a and b:

A linear time algorithm

Two pointers a and b:

- For each prefix maxima s_a

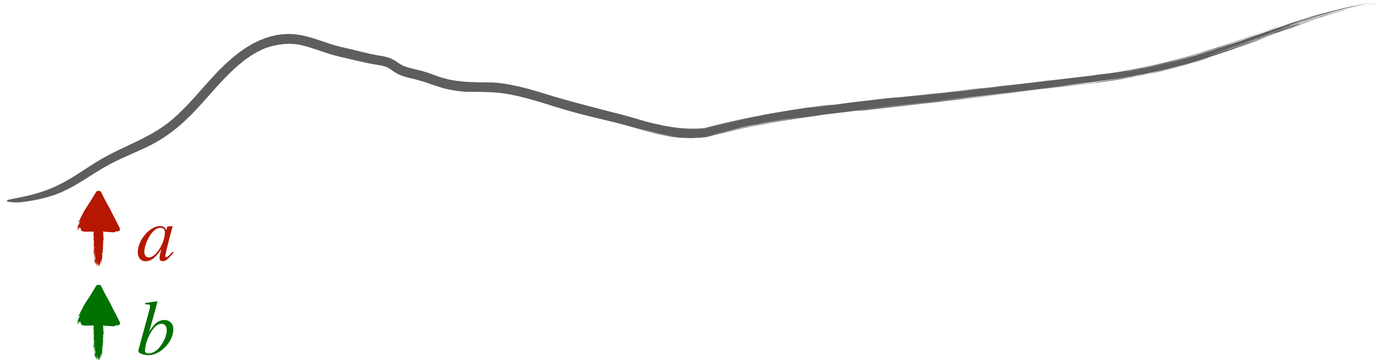
A linear time algorithm



Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**

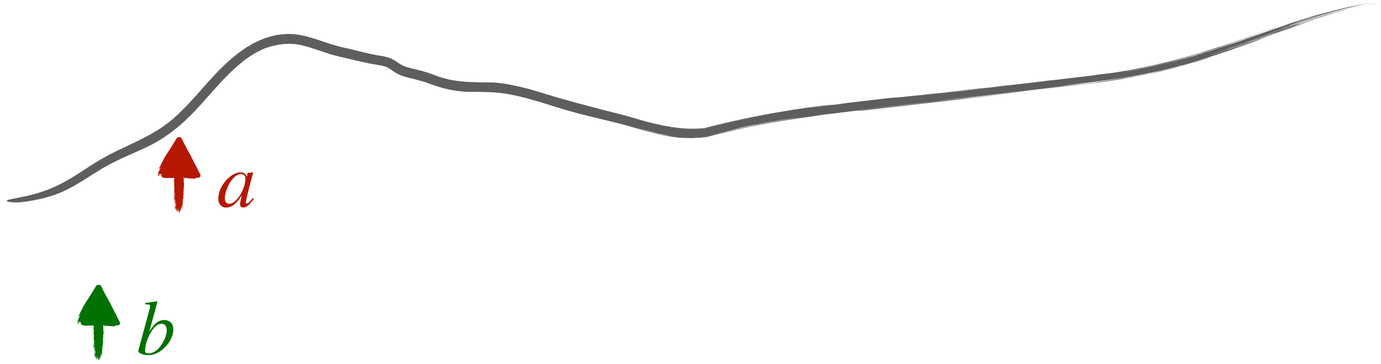
A linear time algorithm



Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**
- Check if $s_a > s_b$
- Check if $s_b \geq \max\{s_1, \ldots, s_{a-1}\}$ **(left maximal)**

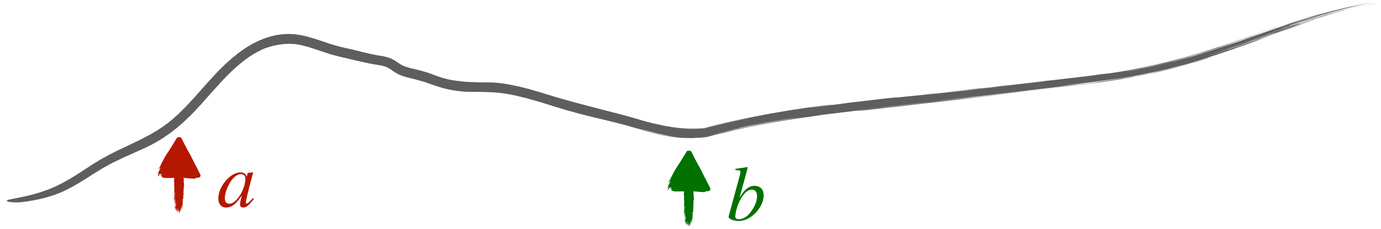
A linear time algorithm



Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**
- Check if $s_a > s_b$
- Check if $s_b \geq \max\{s_1, \ldots, s_{a-1}\}$ **(left maximal)**

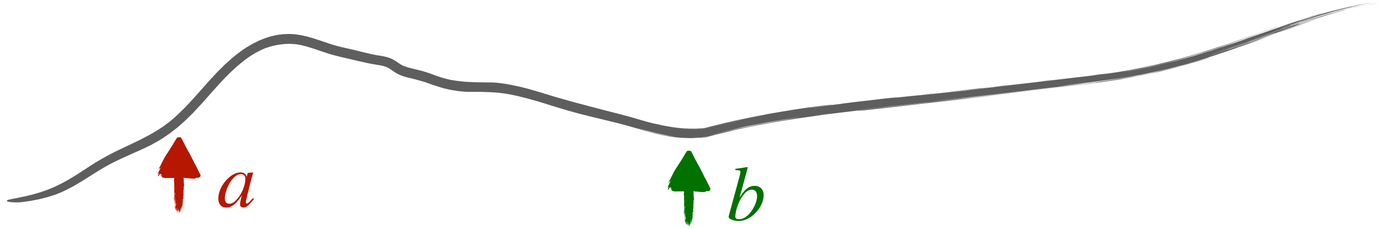
A linear time algorithm



Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**
- Check if $s_a > s_b$
- Check if $s_b \geq \max\{s_1, \ldots, s_{a-1}\}$ **(left maximal)**

A linear time algorithm

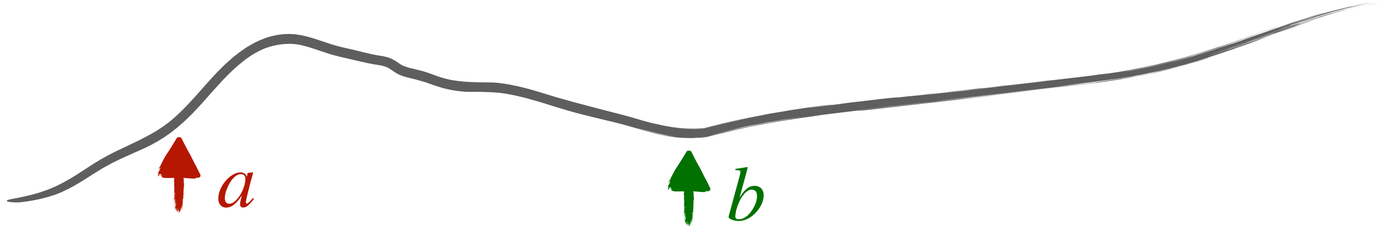


Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**
- Check if $s_a > s_b$
- Check if $s_b \geq \max\{s_1, \ldots, s_{a-1}\}$ **(left maximal)**

} **report** $[a, b)$

A linear time algorithm

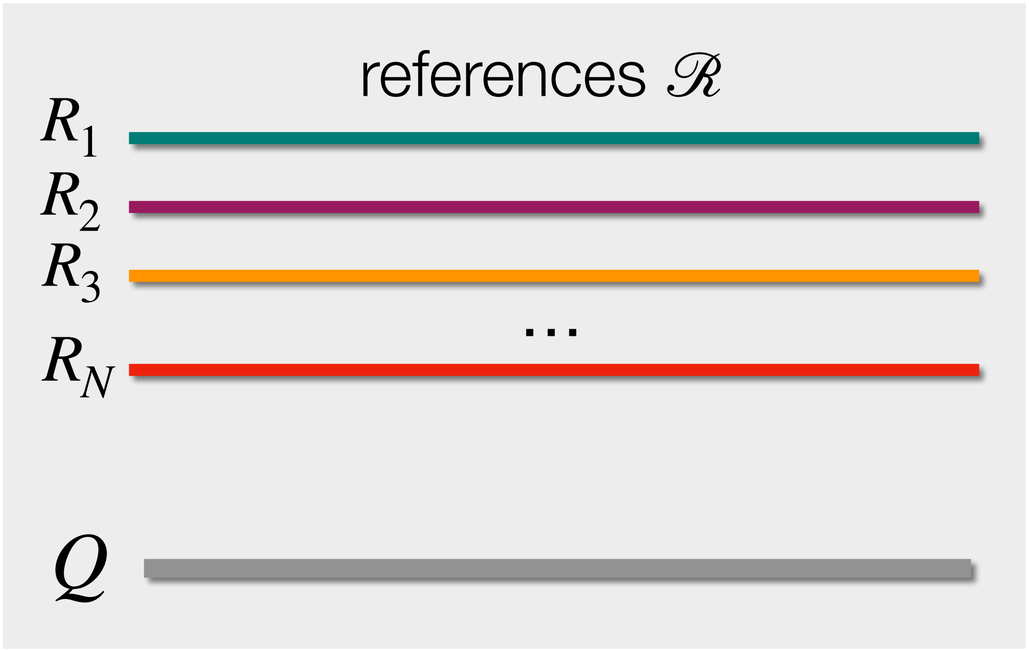


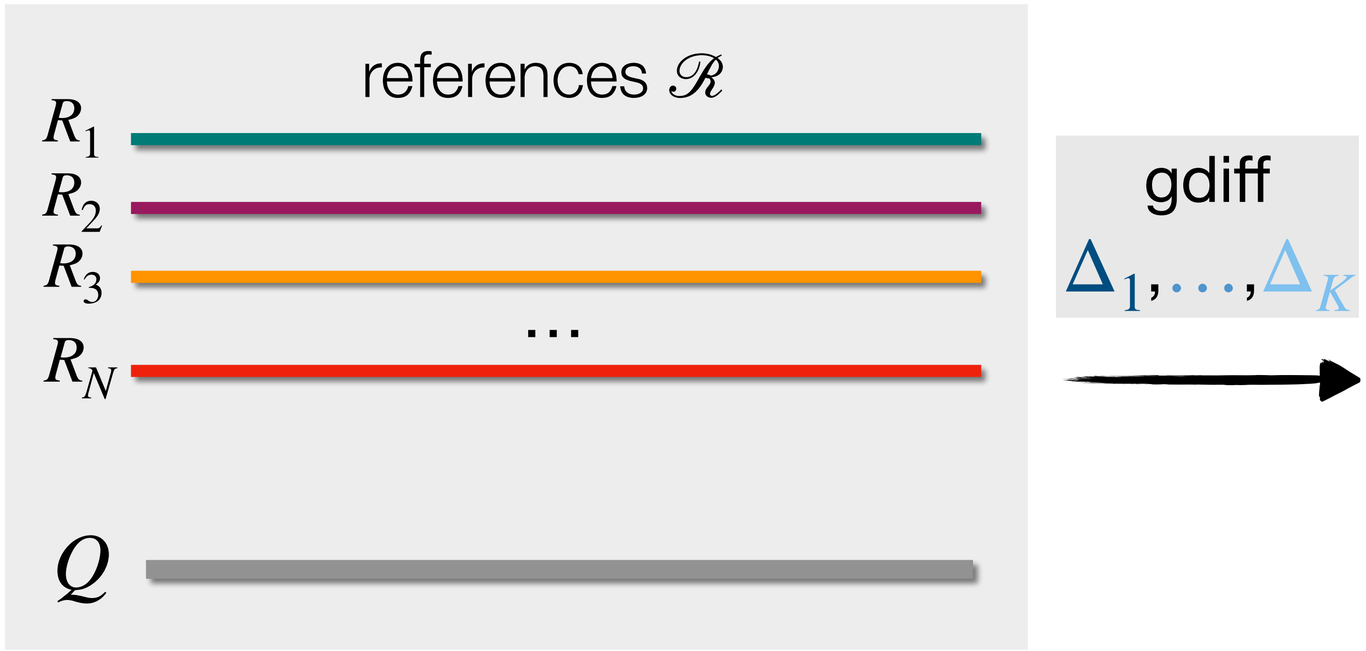
Auxilary arrays for
$\min\{s_{b+1}, \ldots, s_L\}$, $\max\{s_1, \ldots, s_{a-1}\}$

Two pointers a and b:

- For each prefix maxima s_a
- Increment b until $\min\{s_{b+1}, \ldots, s_L\} \geq s_a$ **(right maximal)**
- Check if $s_a > s_b$
- Check if $s_b \geq \max\{s_1, \ldots, s_{a-1}\}$ **(left maximal)**

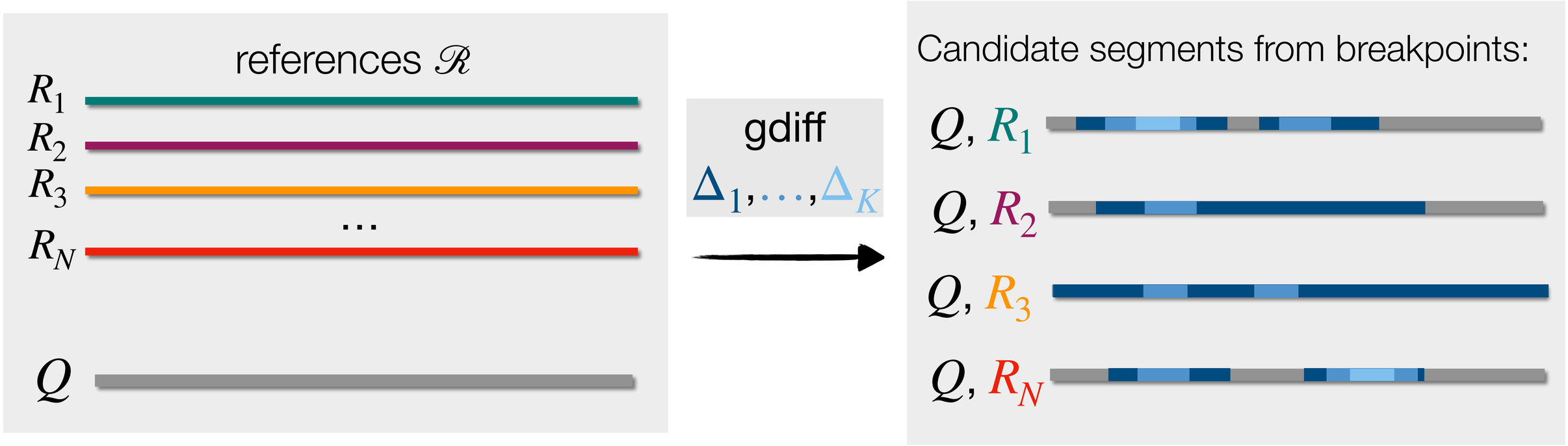
} **report** $[a, b)$

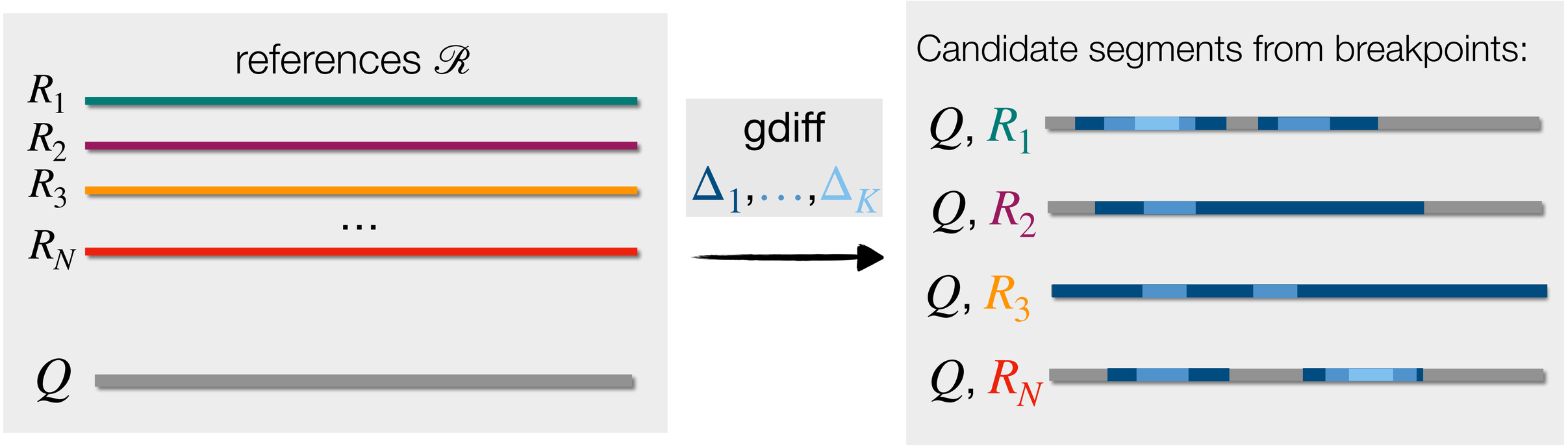

references $\mathscr{R}$
R_1
R_2
R_3
...
R_N
Q



gdiff

$\Delta_1, \ldots, \Delta_K$

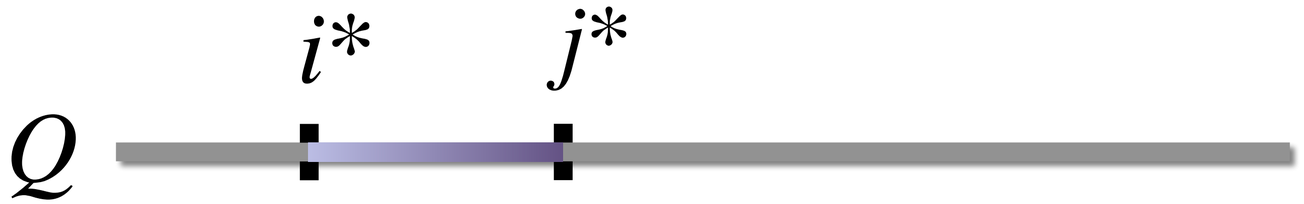

references $\mathscr{R}$
R_1
R_2
R_3
...
R_N
Q
gdiff
$\Delta_1, \ldots, \Delta_K$
Candidate segments from breakpoints:
Q, R_1
Q, R_2
Q, R_3
Q, R_N



~2Mbp genomes, $N = 1000$, $K = 8$, 16 threads → **~12 seconds**

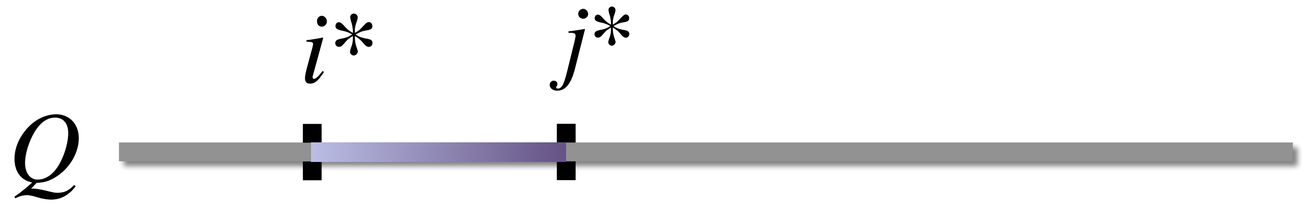
Accounting for rate variation and outliers

Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Accounting for rate variation and outliers

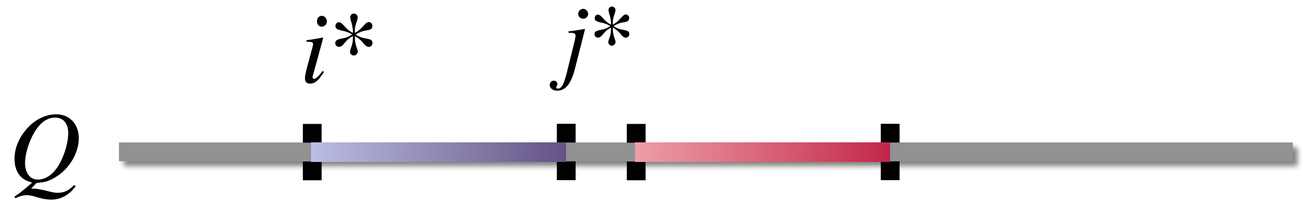
Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Accounting for rate variation and outliers

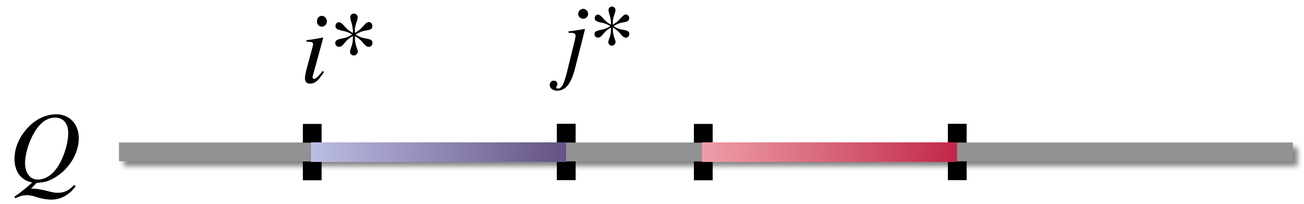
Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Accounting for rate variation and outliers

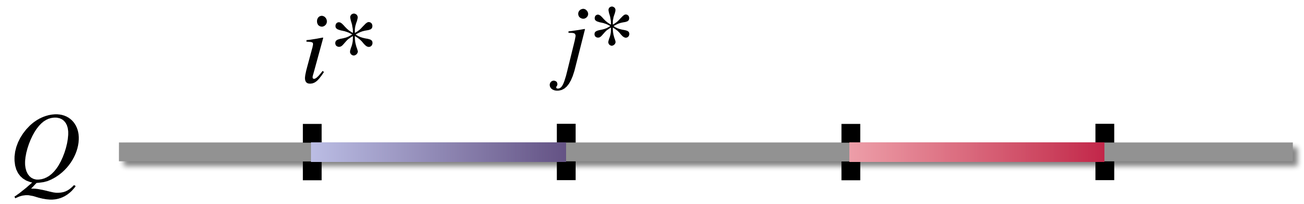
Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Accounting for rate variation and outliers

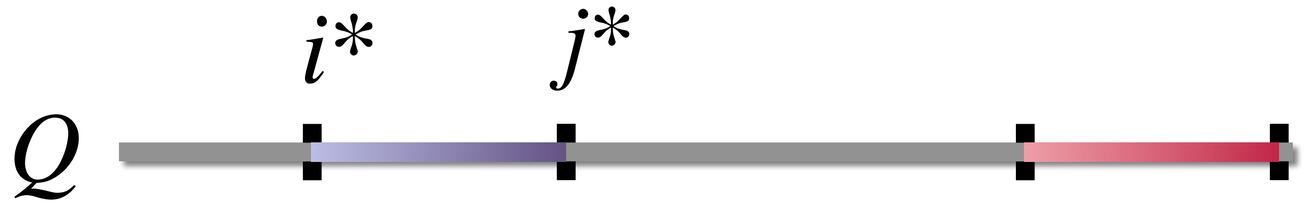
Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Accounting for rate variation and outliers

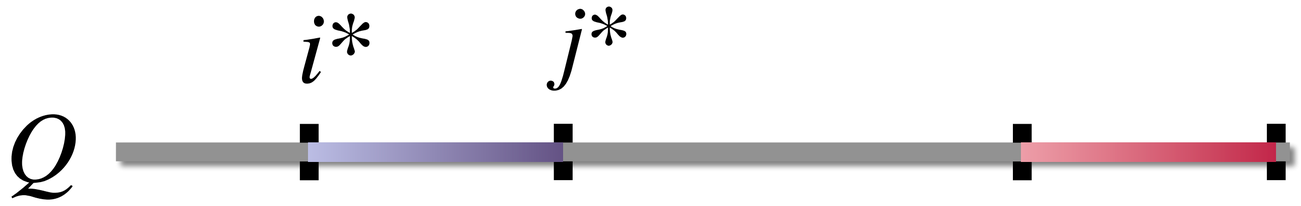
Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

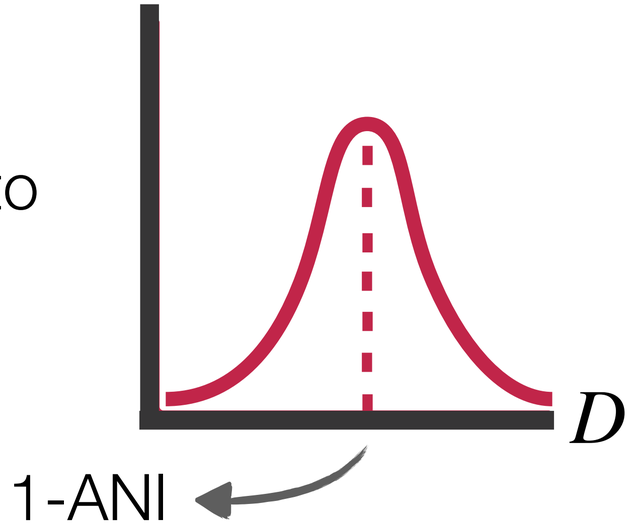
Accounting for rate variation and outliers

Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



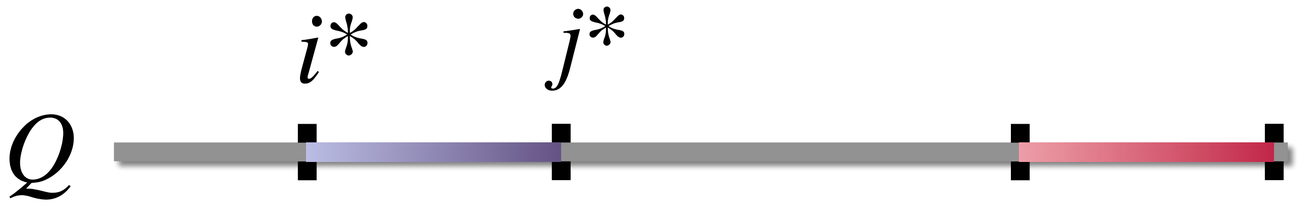
Sample segments across the genome, estimate MLE distances

Fit a Gamma distribution to model the rate variation:



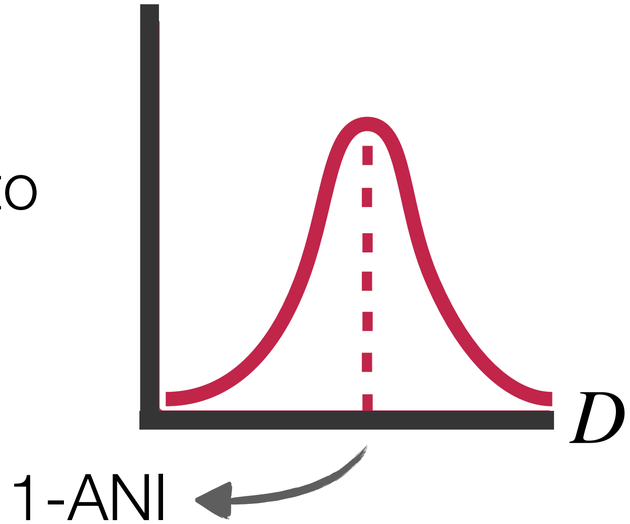
Accounting for rate variation and outliers

Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Fit a Gamma distribution to model the rate variation:

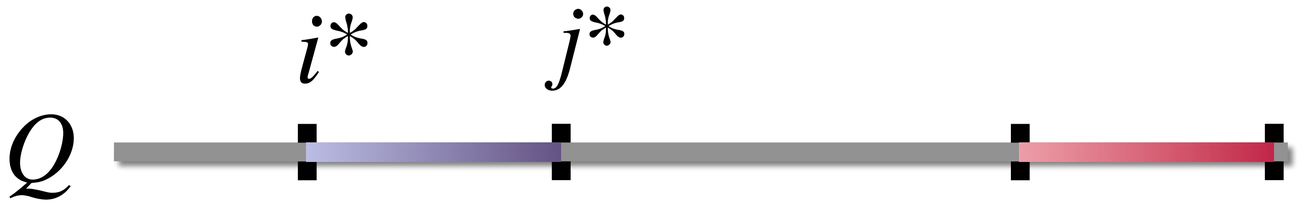


two-sided test for D^*

Obtain a p-value

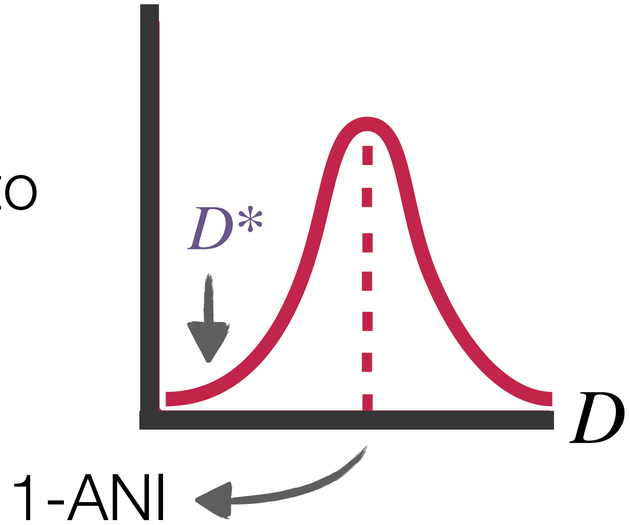
Accounting for rate variation and outliers

Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Fit a Gamma distribution to model the rate variation:



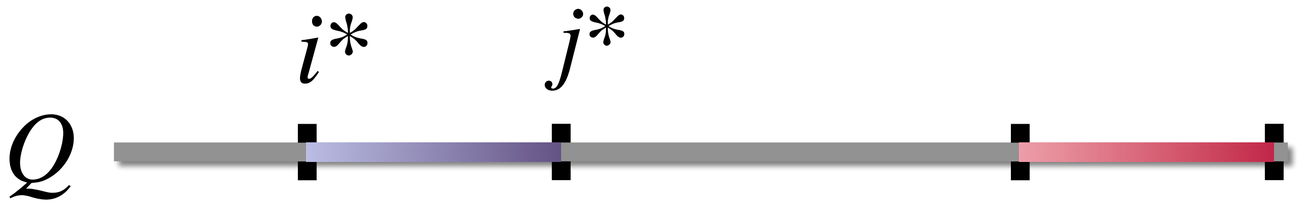
two-sided test for D^*

Obtain a p-value

conserved/HGT?

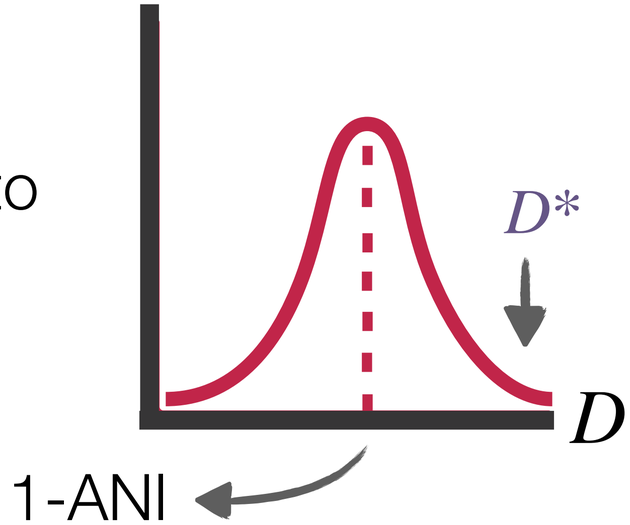
Accounting for rate variation and outliers

Given a candidate segment on Q w.r.t. R, compute its MLE distance D^*



Sample segments across the genome, estimate MLE distances

Fit a Gamma distribution to model the rate variation:



two-sided test for D^*

Obtain a p-value

contamination?

gdiff detects a documented HGT event

Net rate of lateral gene transfer in marine prokaryoplankton

Ramunas Stepanauskas[1,*], Julia M. Brown[1], Shayesteh Arasti[2], Uyen Mai[2], Gregory Gavelis[1], Maria Pachiadaki[3], Oliver Bezuidt[1,4], Jacob H. Munson-McGee[1], Tianyi Chang[1], Steven J. Biller[5], Paul M. Berube[6], Siavash Mirarab[7]

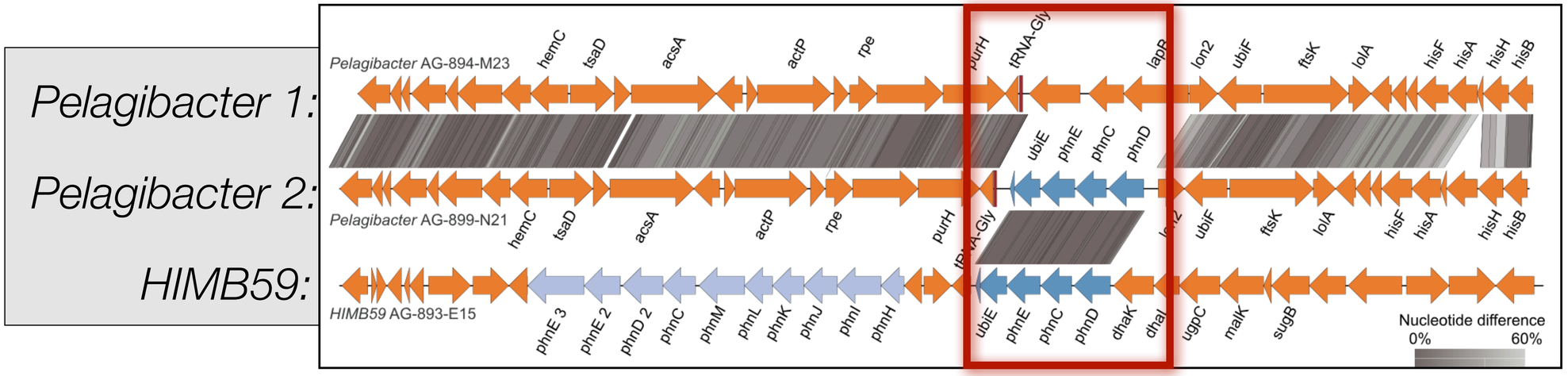
- A horizontally transferred gene (1.6% gene distance) between a *Pelagibacter* & a ***Proteobacterium HIMB59*** (29.4% genome-wide distance).

gdiff detects a documented HGT event

Net rate of lateral gene transfer in marine prokaryoplankton

Ramunas Stepanauskas[1,*], Julia M. Brown[1], Shayesteh Arasti[2], Uyen Mai[2], Gregory Gavelis[1], Maria Pachiadaki[3], Oliver Bezuidt[1,4], Jacob H. Munson-McGee[1], Tianyi Chang[1], Steven J. Biller[5], Paul M. Berube[6], Siavash Mirarab[7]

- A horizontally transferred gene (1.6% gene distance) between a *Pelagibacter* & a ***Proteobacterium HIMB59*** (29.4% genome-wide distance).

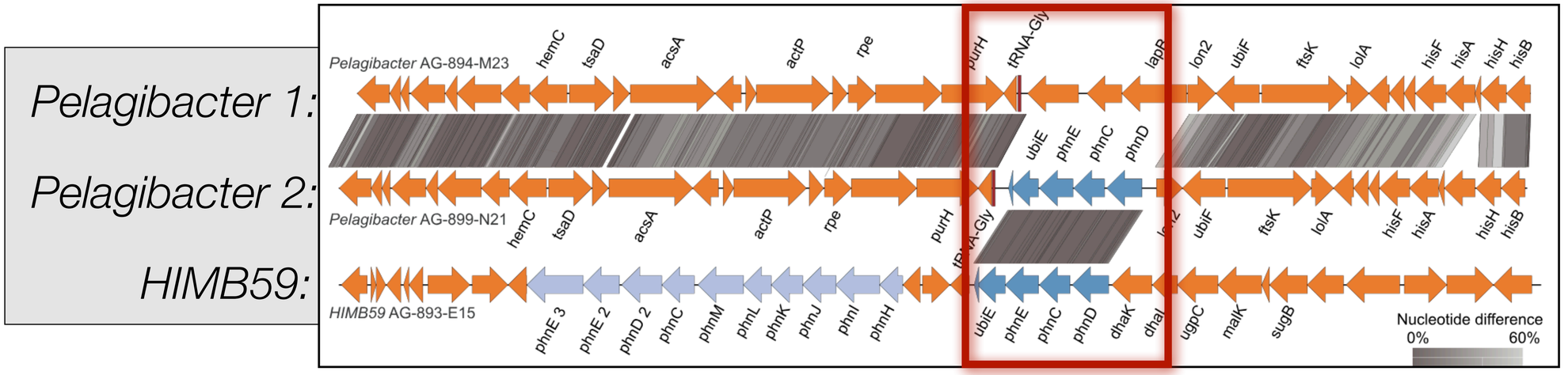


gdiff detects a documented HGT event

Net rate of lateral gene transfer in marine prokaryoplankton

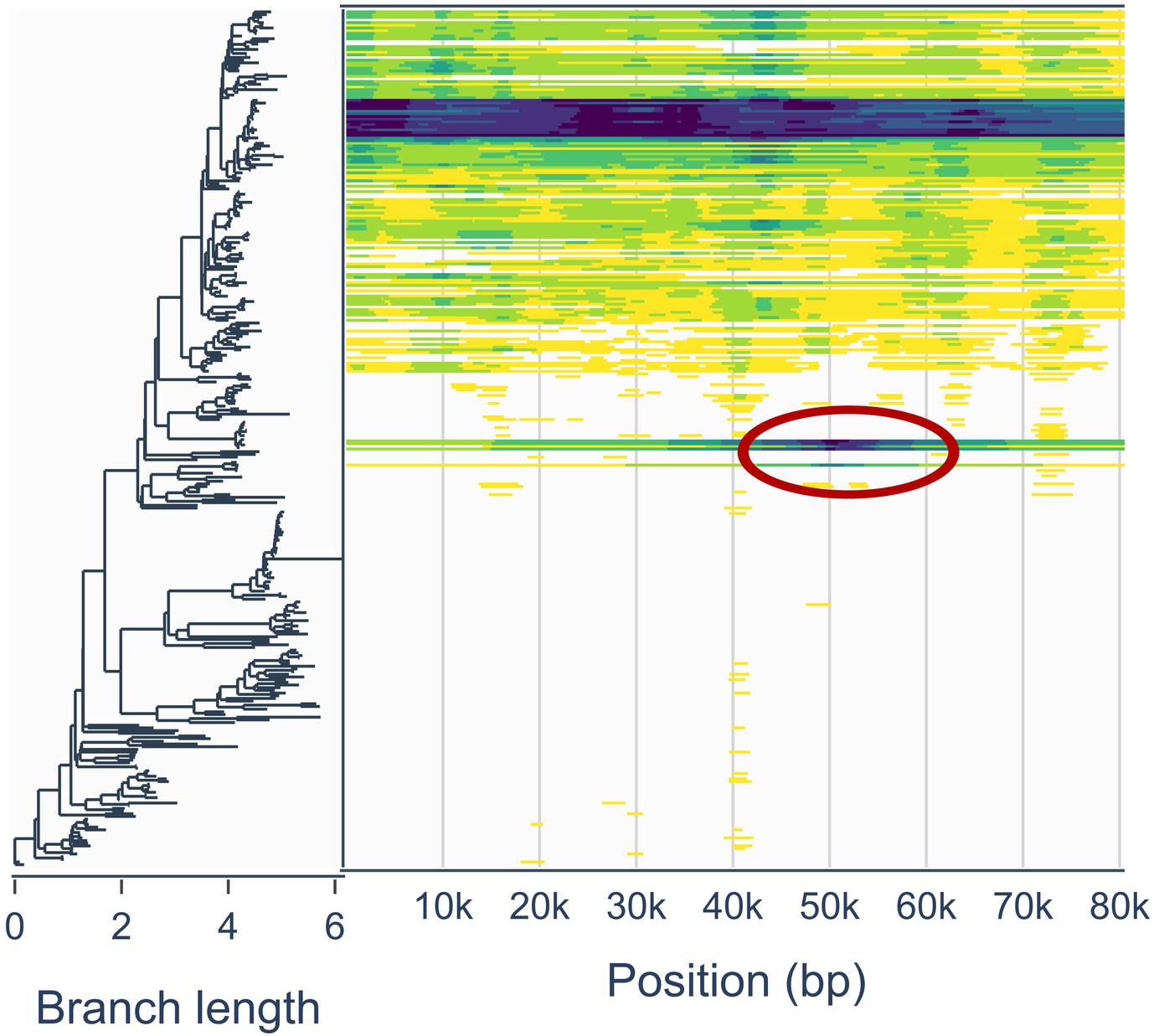
Ramunas Stepanauskas[1,*], Julia M. Brown[1], Shayesteh Arasti[2], Uyen Mai[2], Gregory Gavelis[1], Maria Pachiadaki[3], Oliver Bezuidt[1,4], Jacob H. Munson-McGee[1], Tianyi Chang[1], Steven J. Biller[5], Paul M. Berube[6], Siavash Mirarab[7]

- A horizontally transferred gene (1.6% gene distance) between a *Pelagibacter* & a ***Proteobacterium HIMB59*** (29.4% genome-wide distance).
- Querying a single-cell assembled (SAG) ***Proteobacterium HIMB59*** against 10,000 marine SAG references → scalable to tens of thousands of genomes!



A documented HGT event in ocean microbes

500 genomes selected from GORG-Tropical dataset (10,000)



Summary: gdiff

A framework for local distance estimation & genomic “outlier” detection

- scalable, based on “homologous” *k*-mers, models rate variation
- detecting contaminations, conserved regions, viral integration, HGT

Summary: gdiff

A framework for local distance estimation & genomic “outlier” detection

- scalable, based on “homologous” *k*-mers, models rate variation
- detecting contaminations, conserved regions, viral integration, HGT

Future work includes

- going beyond pairwise comparisons & adding the phylogenetic aspect

Summary: gdiff

A framework for local distance estimation & genomic “outlier” detection

- scalable, based on “homologous” *k*-mers, models rate variation
- detecting contaminations, conserved regions, viral integration, HGT

Future work includes

- going beyond pairwise comparisons & adding the phylogenetic aspect
- classifying detected regions & identifying the type of the outlier

Summary: gdiff

A framework for local distance estimation & genomic “outlier” detection

- scalable, based on “homologous” *k*-mers, models rate variation
- detecting contaminations, conserved regions, viral integration, HGT

Future work includes

- going beyond pairwise comparisons & adding the phylogenetic aspect
- classifying detected regions & identifying the type of the outlier
- better benchmarking & application to other datasets (e.g., eukaryotes)

Thank you!

Siavash Mirarab

Eduardo Charvel

Funding

MINDEROO FOUNDATION

NIH

software: github.com/bo1929/gdiff

paper:

Work in progress