Conditional random fields

Conditional random fields

- **Definition**

$$P(\pi \mid x) = \frac{\exp(\Sigma_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}{\Sigma_{\pi'} \exp(\Sigma_{i=1 \ldots |x|} w^T F(\pi'_i, \pi'_{i-1}, x, i))}$$

partition coefficient

where

F : (state, state, observations, index) → $\mathbf{R}^n$ "local feature mapping"

w ∈ $\mathbf{R}^n$ "parameter vector"

- Summation over all possible state sequences $\pi'_1 \ldots \pi'_{|x|}$
- $a^T b$ for vectors a, b ∈ $\mathbf{R}^n$ denotes inner product, $\Sigma_{i=1 \ldots n}\, a_i\, b_i$

Relationship with HMMs

$\log P(x, \pi) = \log a_0(\pi_0) + \Sigma_{i=1 \ldots |x|} [\log a(\pi_{i-1}, \pi_i) + \log e_{\pi i}(x_i)]$ (*)

- For each component w_j, define F_j to be a 0/1 indicator variable of whether the jth parameter should be included in scoring x, π at position i:

$$w = \begin{bmatrix} \log a_0(1) \\ \ldots \\ \log a_0(K) \\ \log a_{11} \\ \ldots \\ \log a_{KK} \\ \log e_1(b_1) \\ \ldots \\ \log e_K(b_M) \end{bmatrix} \in \mathbf{R}^n \qquad F(\pi_i, \pi_{i-1}, x, i) = \begin{bmatrix} 1\{i = 1 \wedge \pi_{i-1} = 1\} \\ \ldots \\ 1\{i = 1 \wedge \pi_{i-1} = K\} \\ 1\{\pi_{i-1} = 1 \wedge \pi_i = 1\} \\ \ldots \\ 1\{\pi_{i-1} = K \wedge \pi_i = K\} \\ 1\{x_i = b_1 \wedge \pi_i = 1\} \\ \ldots \\ 1\{x_i = b_M \wedge \pi_i = K\} \end{bmatrix} \in \mathbf{R}^n$$

- Then, $\log P(x, \pi) = \Sigma_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)$

Relationship with HMMs

$$\log P(x, \pi) = \Sigma_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)$$

- Equivalently,

$$P(\pi \mid x) = \frac{P(x, \pi)}{\Sigma_\pi P(x, \pi)} = \frac{\exp(\Sigma_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}{\Sigma_{\pi'} \exp(\Sigma_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)}$$

- Therefore, an HMM can be converted to an equivalent CRF

CRFS ≥ HMMs (continued)

- In an HMM, our features were of the form

$$F(\pi_i, \pi_{i-1}, x, i) = F(\pi_i, \pi_{i-1}, x_i, i)$$

  - I.e., when scoring position i in the sequence, feature only considered the emission x_i at position i.
  - Cannot look at other positions (e.g., x_{i-1}, x_{i+1}) since that would involve “emitting” a character more than once – double-counting of probability

- CRFs don’t have this restriction
  - Why? Because CRFs don’t attempt to model the observations x!

Examples of non-local features for CRFs

- Casino:
  - Dealer looks at previous 100 positions, and determines whether at least 50 over them had 6's

    $F_j(\text{LOADED}, \text{FAIR}, x, i) = 1\{ x_{i-100} \ldots x_i \text{ has} > 50 \text{ 6s} \}$

- CpG islands:
  - Gene occurs near a CpG island

    $F_j(*, \text{EXON}, x, i) = 1\{ x_{i-1000} \ldots x_{i+1000} \text{ has} > 1/16 \text{ CpGs} \}$

3 basic questions for CRFs

- **Evaluation:** Given a sequence of observations x and a sequence of states π, compute P(π | x)

- **Decoding:** Given a sequence of observations x, compute the maximum probability sequence of states $\pi_{ML} = \arg\max_{\pi} P(\pi \mid x)$

- **Learning:** Given a CRF with unspecified parameters w, compute the parameters that maximize the likelihood of π given x, i.e., $w_{ML} = \arg\max_{w} P(\pi \mid x, w)$

Viterbi for CRFs

- Note that:

$$\arg\max_{\pi} P(\pi \mid x) = \arg\max_{\pi} \frac{\exp(\Sigma_{i=1 \ldots |x|}\, w^T F(\pi_i, \pi_{i-1}, x, i))}{\Sigma_{\pi'} \exp(\Sigma_{i=1 \ldots |x|}\, w^T F(\pi'_i, \pi'_{i-1}, x, i))}$$

$$= \arg\max_{\pi} \exp(\Sigma_{i=1 \ldots |x|}\, w^T F(\pi_i, \pi_{i-1}, x, i))$$

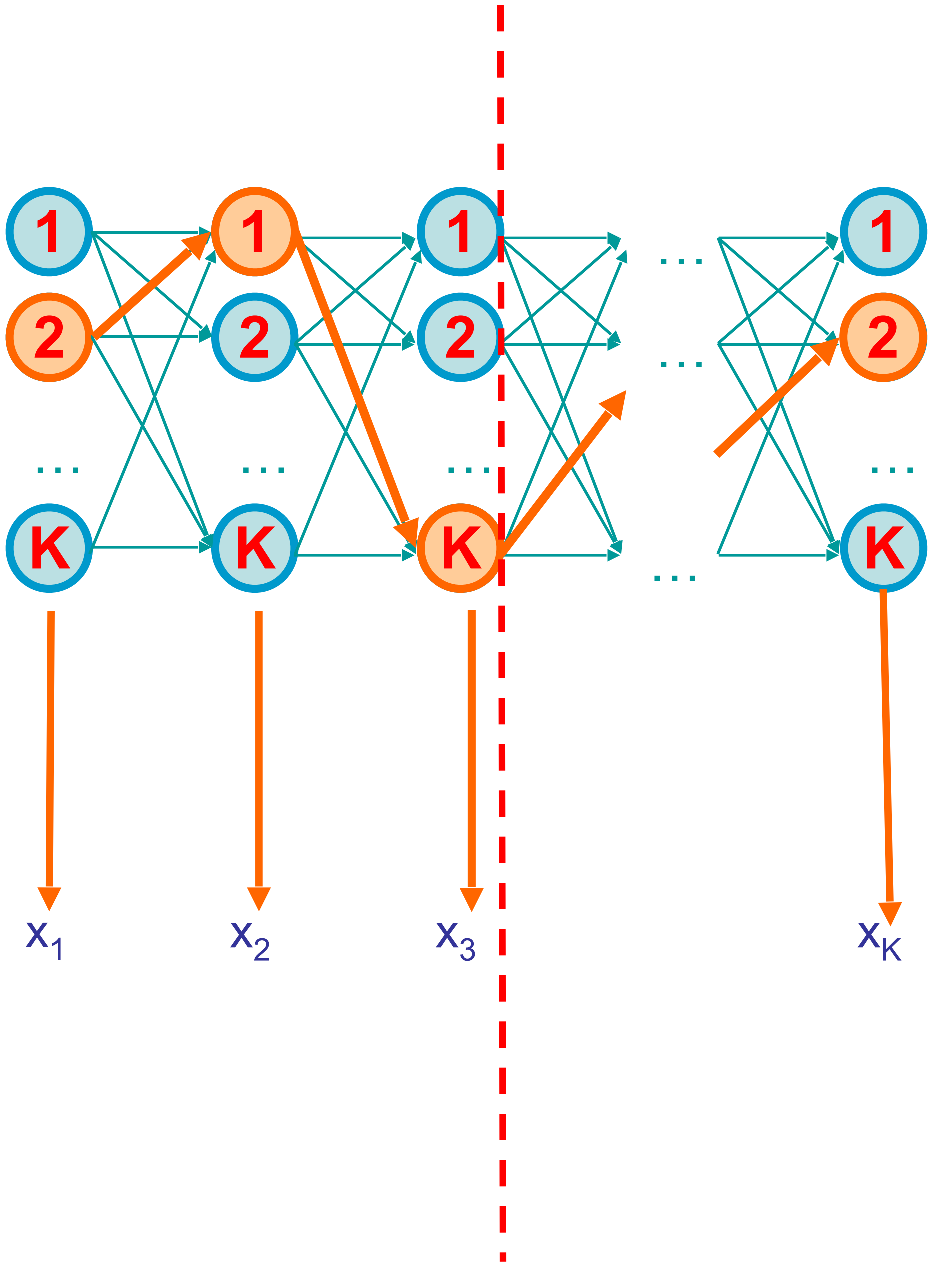
$$= \arg\max_{\pi} \Sigma_{i=1 \ldots |x|}\, w^T F(\pi_i, \pi_{i-1}, x, i)$$

- We can derive the following recurrence:

$$V_k(i) = \max_j [\, w^T F(k, j, x, i) + V_j(i-1) \,]$$

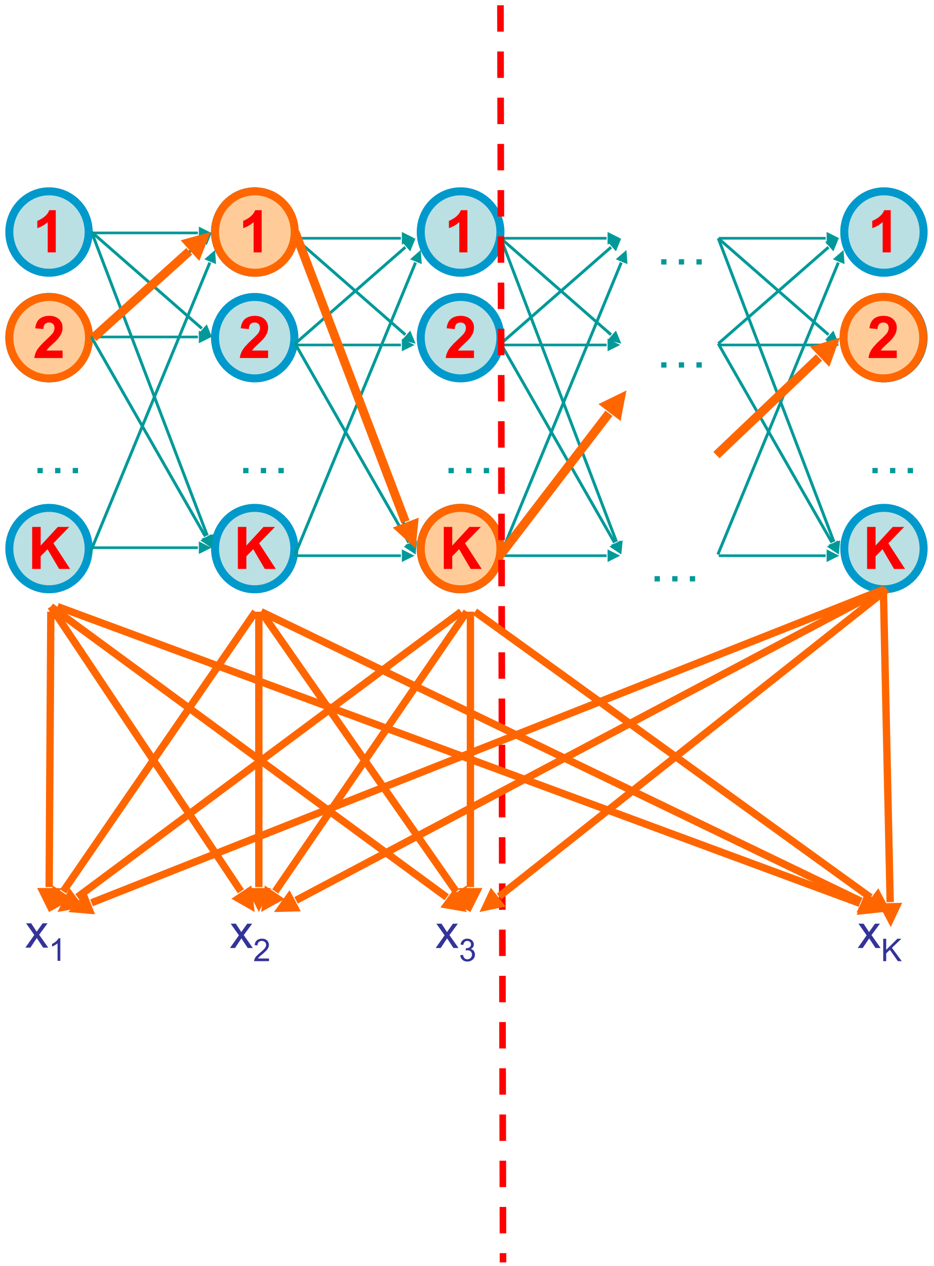
- **Notes:**
  - Even though the features may depend on arbitrary positions in x, x is constant. DP depends only on knowing the previous state
  - Computing the partition function (denominator) can be done by a similar adaptation of the forward/backward algorithms

Viterbi for CRFs



Given that we end up in state k at step i, maximize score to the left and right

Viterbi for CRFs

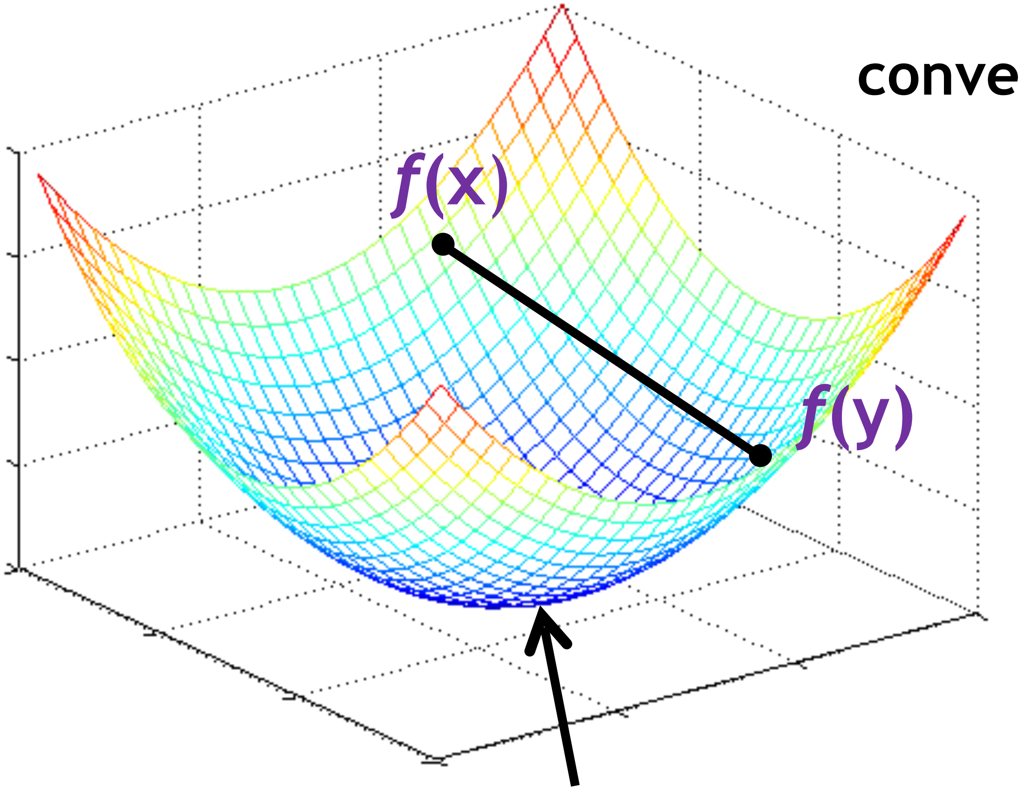


Given that we end up in state k at step i, maximize score to the left and right

X is fixed:
=> parse to the left of step i, given we end in state k, does not affect parse to the right of step i

Learning CRFs

- Key observation: $-\log P(\pi \mid x, w)$ is a differentiable, **convex** function of w

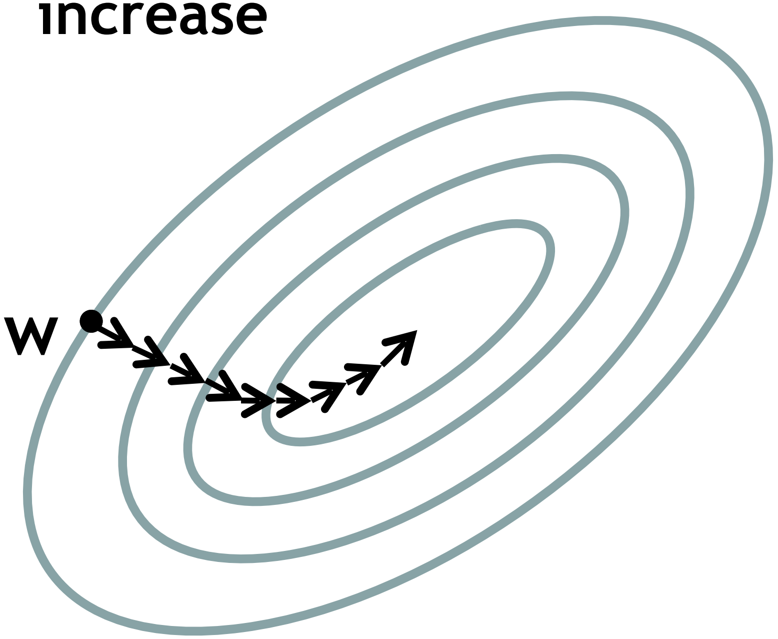


convex function

Any local minimum is a global minimum.

Learning CRFs (continued)

- Compute partial derivative of log P(π | x, w) with respect to each parameter w_j, and use the gradient ascent learning rule:



Gradient points in the direction of greatest function increase

The CRF gradient

- It turns out that

$$(\partial/\partial w_j) \log P(\pi \mid x, w) = F_j(x, \pi) - E_{\pi' \sim P(\pi' \mid x, w)} [\, F_j(x, \pi') \,]$$

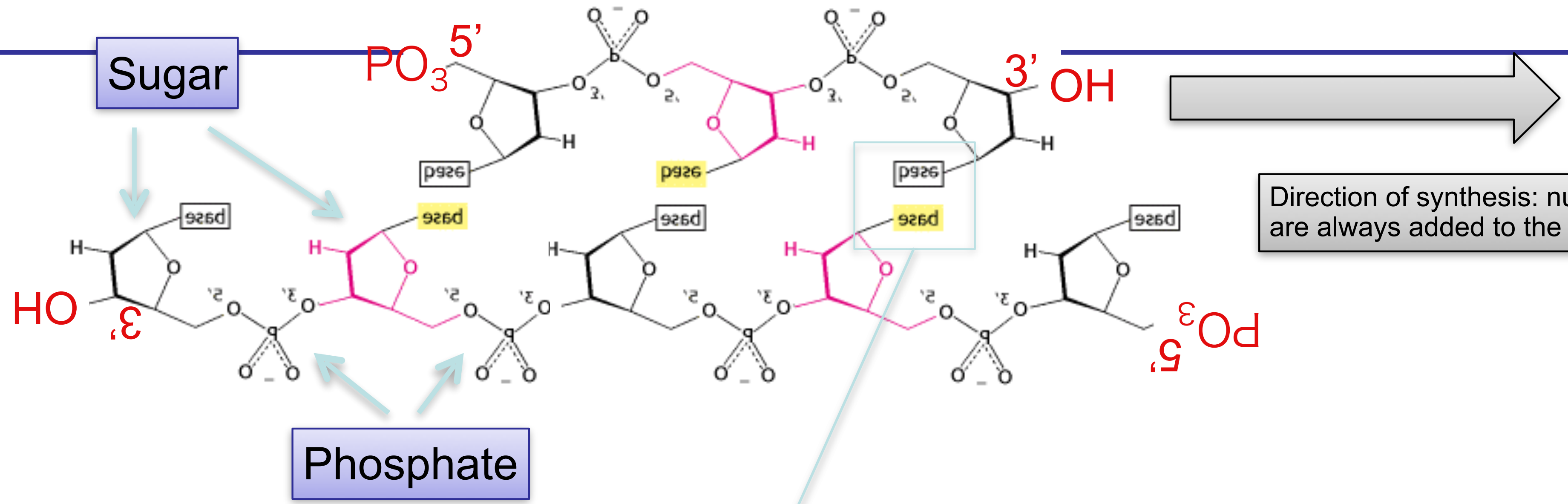
correct value for jth feature

expected value for jth feature (given the current parameters)

- This has a very nice interpretation:
  - We increase parameters for which the correct feature values are greater than the predicted feature values
  - We decrease parameters for which the correct feature values are less than the predicted feature values
- This moves probability mass from incorrect parses to correct parses

DNA Structure

DNA structure

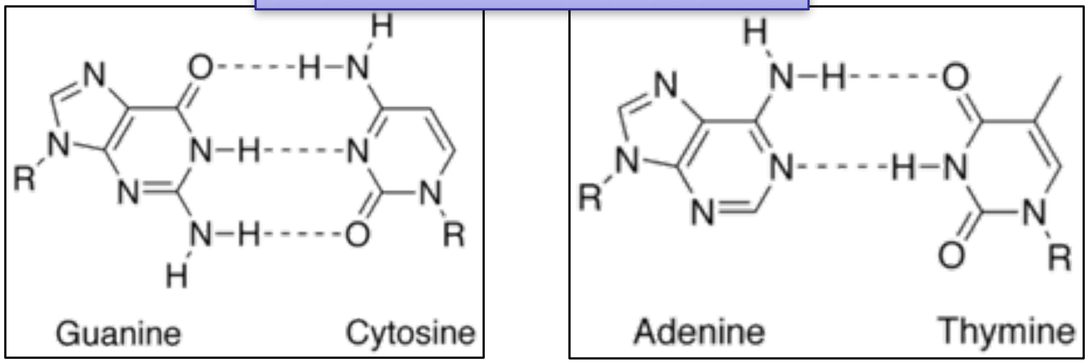


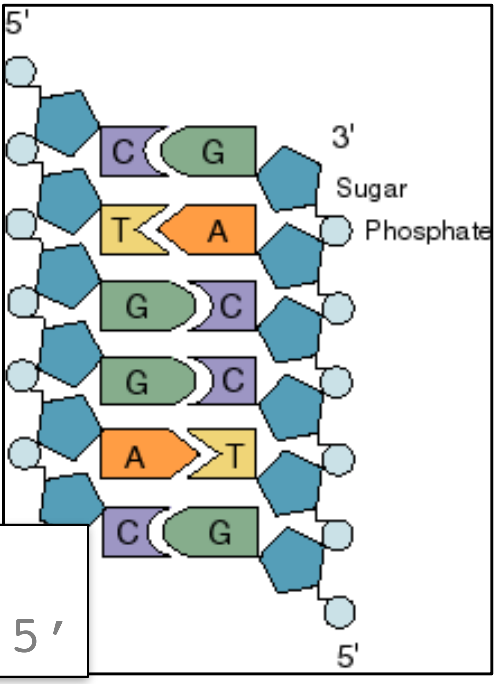
Sugar

Phosphate

Direction of synthesis: nucleotides are always added to the 3' end.

Base pairs: G-C and A-T



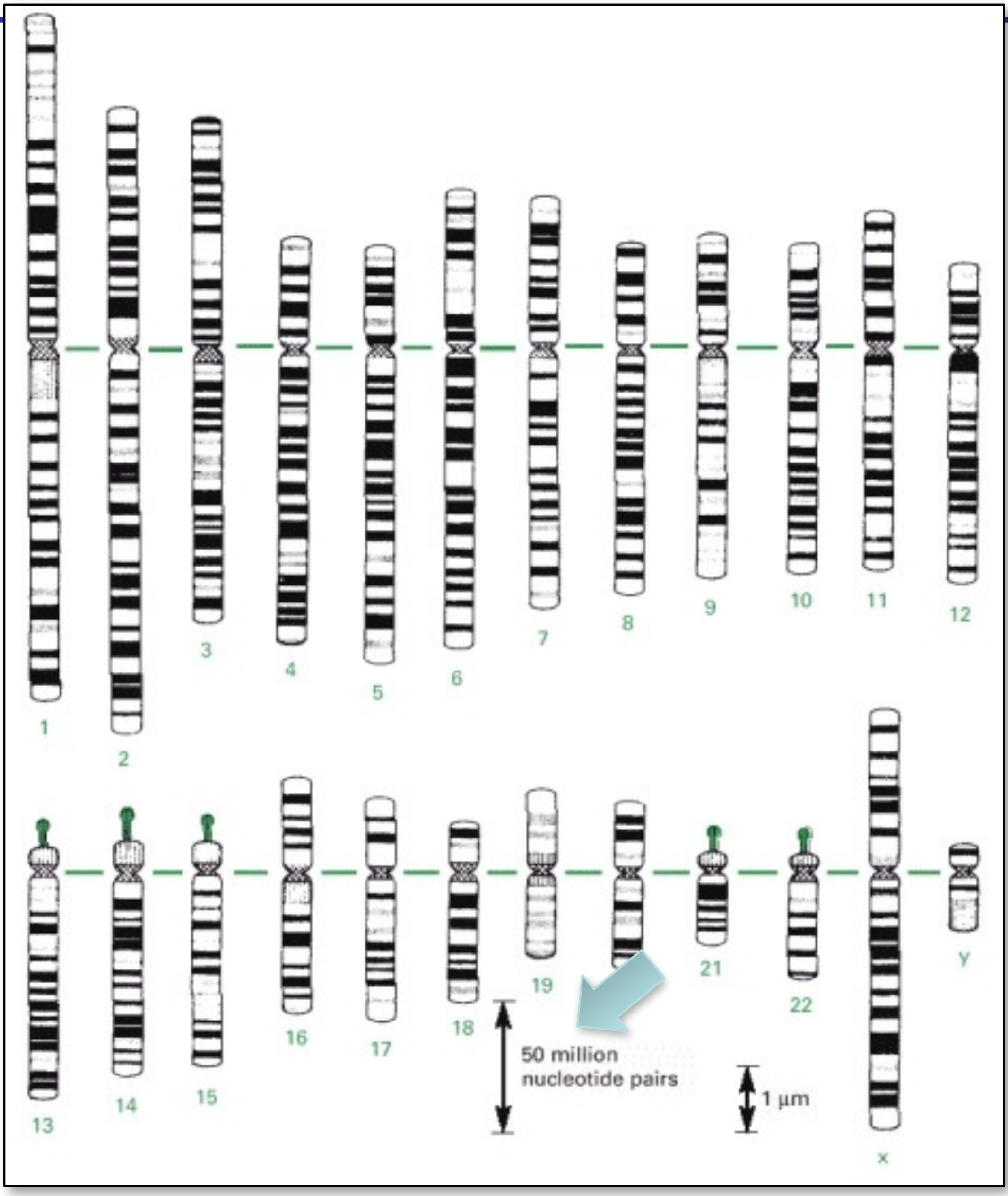


We only write out Watson!

```
Watson  5'   C T G G A C   3'
Crick          3'   G A C C T G   5'
```

Slide Credit: Arend Sidow

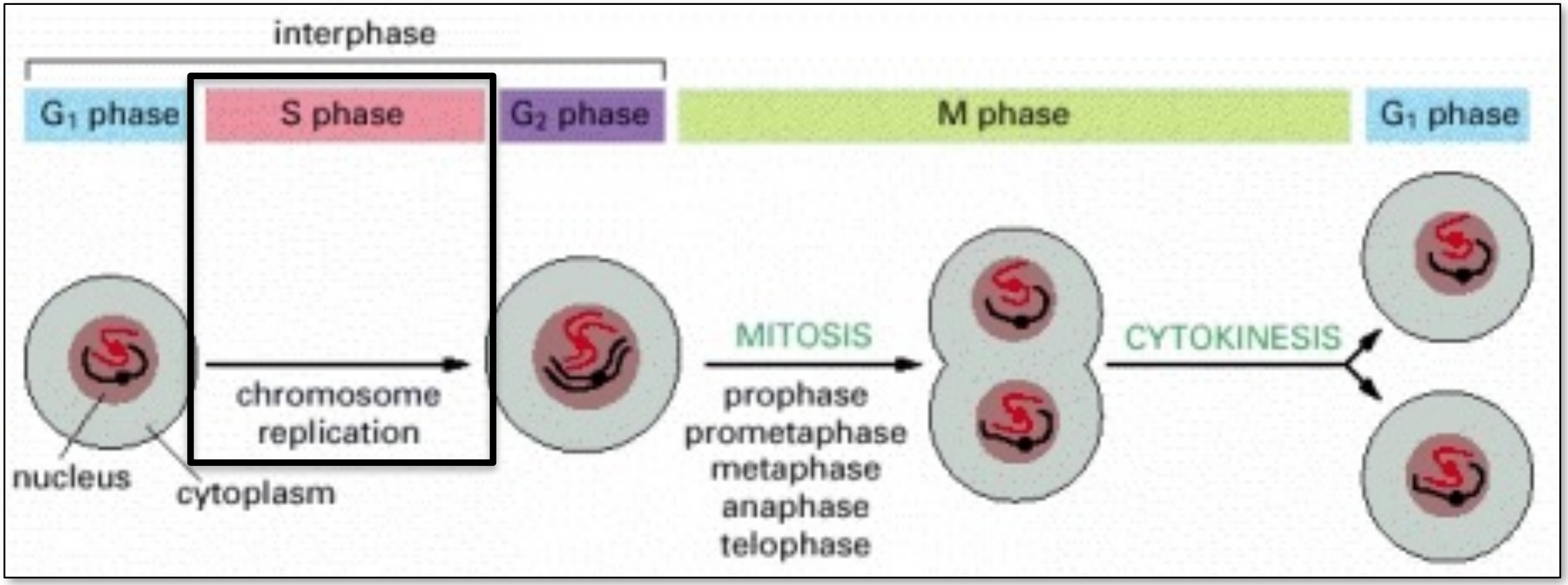
Human chromosomes



- 3,000 million base pairs total
- One replication origin every ~50 kb
- Replication happens only during a short specific period

Slide Credit: Arend Sidow

Cell cycle



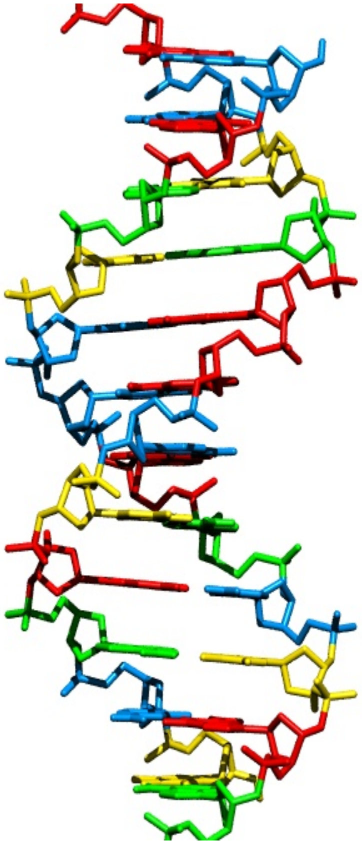
- DNA replication happens during a short time period
- Except in very early nonmammalian embryos, most time is spent in G1 doing useful stuff
- Even in cancer cells, most time is spent in G1 because cells don’t divide until the daughter cells have grown back to standard cell size, and that requires lots of transcription and protein synthesis.

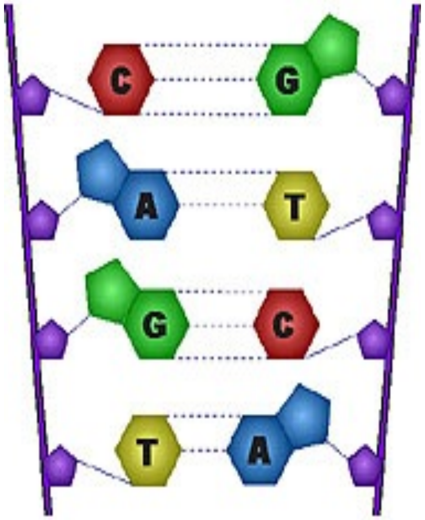
Slide Credit: Arend Sidow

DNA Sequencing

DNA sequencing

How we obtain the sequence of nucleotides of a species





…ACGTGACTGAGGACCGTG

CGACTGAGACTGACTGGGT

CTAGCTAGACTACGTTTTA

TATATATATACGTCGTCGT

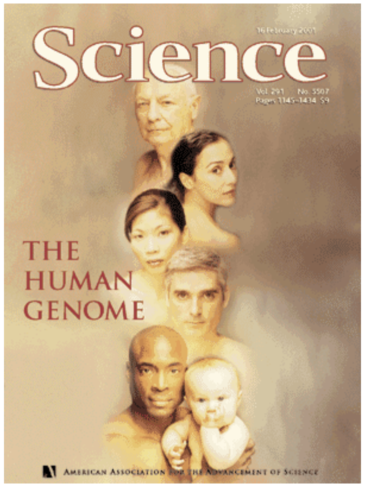
ACTGATGACTAGATTACAG

ACTGATTTAGATACCTGAC

TGATTTTAAAAAAATATT…

Human Genome Project

nature
the human genome
Nuclear fission
Five-dimensional energy landscapes
Seafloor spreading
The view from under the Arctic ice
Career prospects
Sequence creates new opportunities
naturejobs
genomics special



1990: Start

2000: Bill Clinton:

“most important scientific discovery in the 20th century”

2001: Draft

2003: Finished

3 billion basepairs

$3 billion

now what?

Which representative of the species?

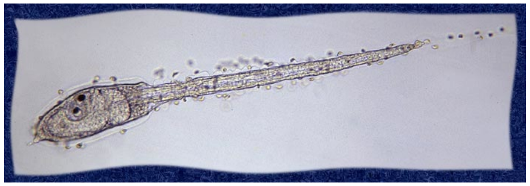
Which human?

Answer one:

Answer two: it doesn't matter

Polymorphism rate: number of letter changes between two different members of a species

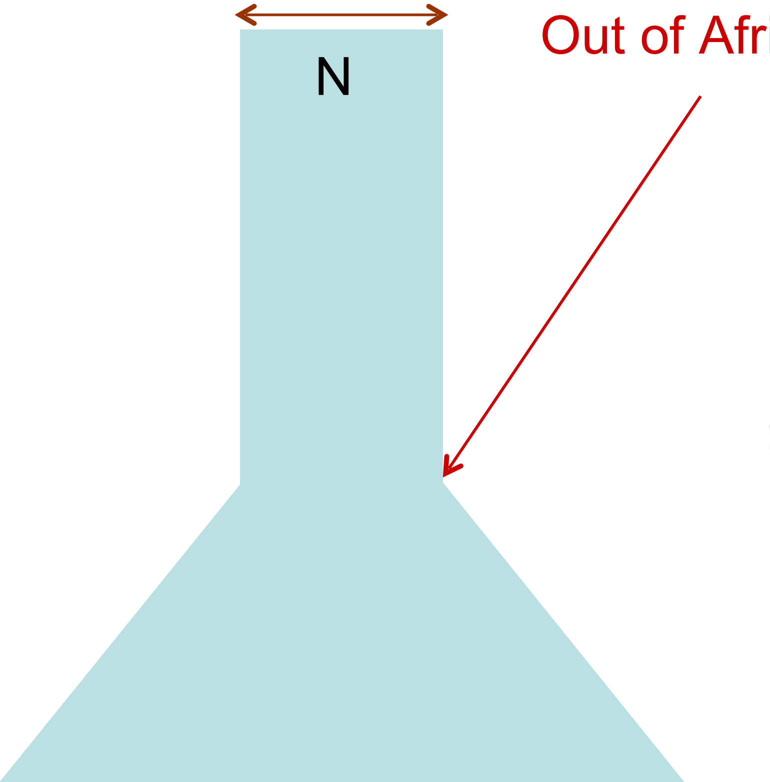
Humans: ~1/1,000

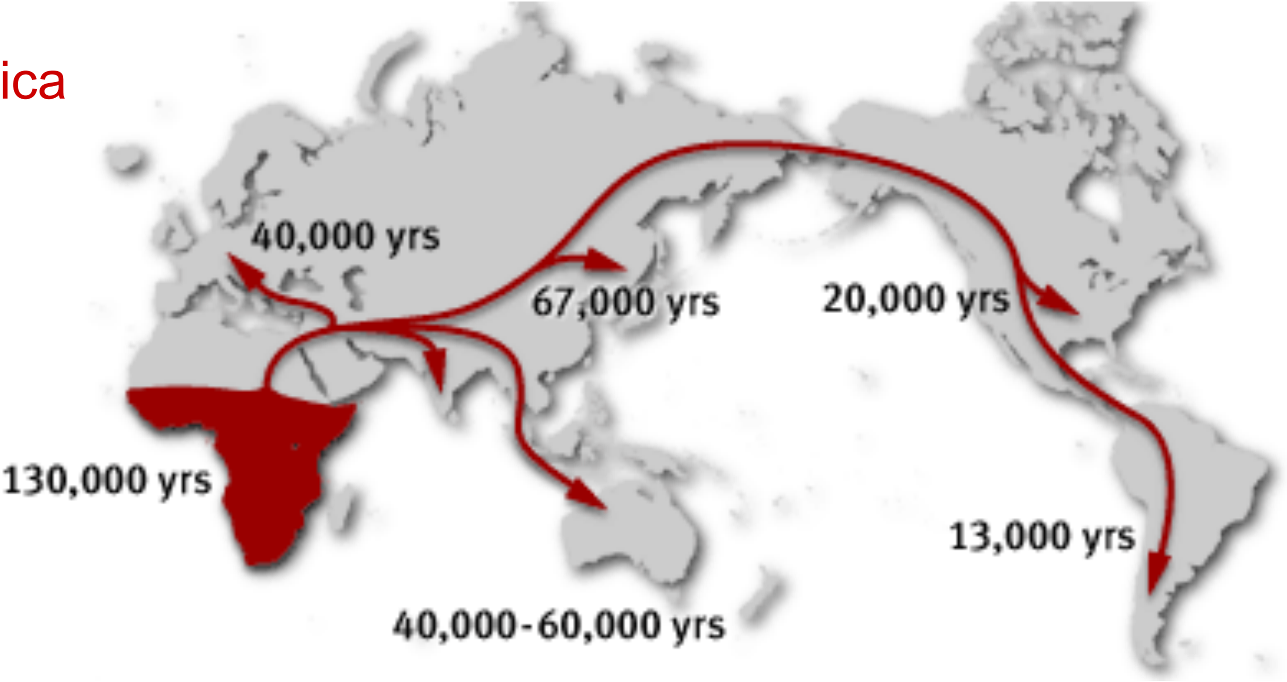


Other organisms have much higher polymorphism rates

- Population size!

Why humans are so similar





A small population that interbred reduced the genetic variation

Out of Africa ~ 40,000 years ago

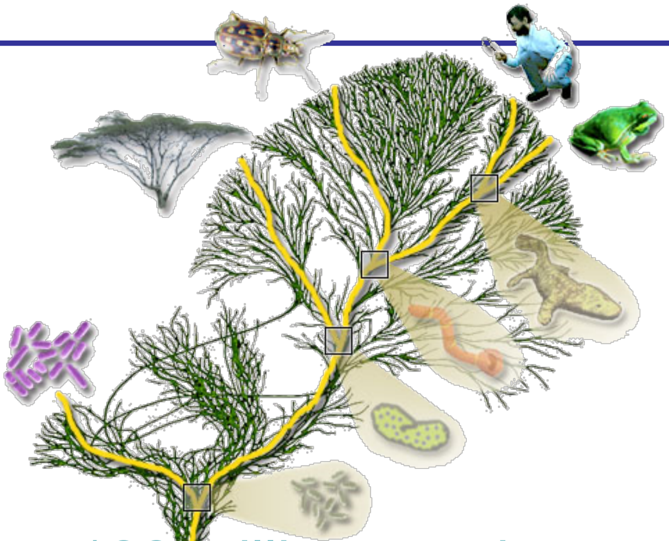
Heterozygosity: H

$H = 4Nu/(1 + 4Nu)$

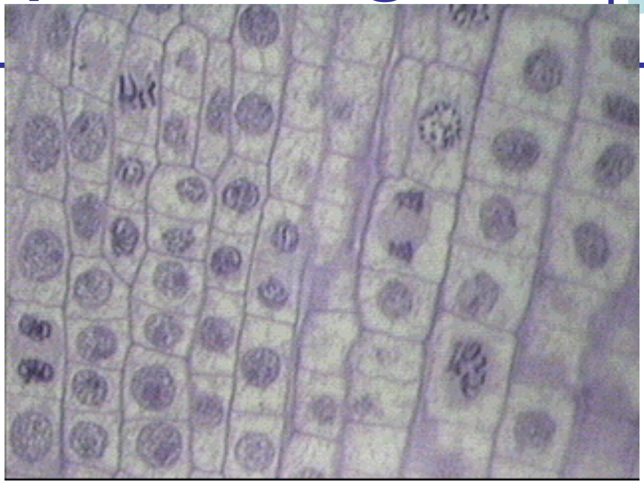
$u \sim 10^{-8}$, $N \sim 10^{4}$

$\Rightarrow H \sim 4 \times 10^{-4}$

There is never "enough" sequencing

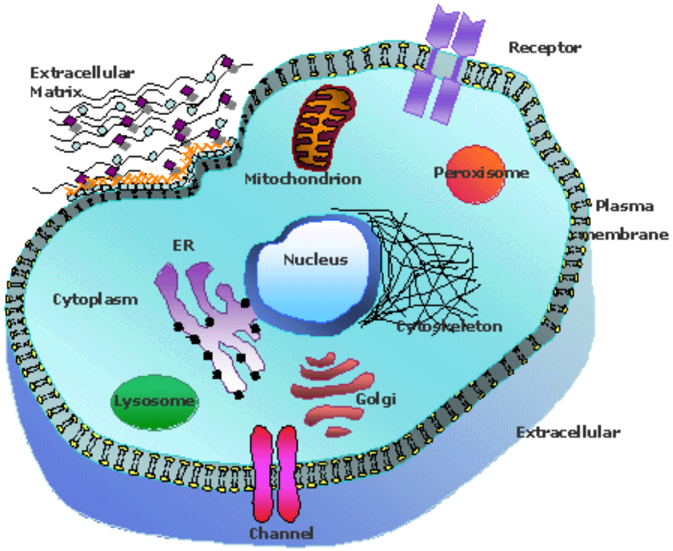


100 million species



Somatic mutations
(e.g., HIV, cancer)

7 billion individuals

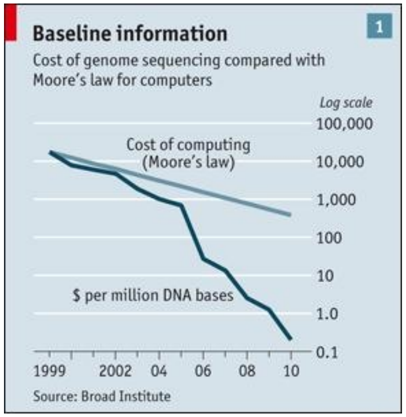


Sequencing is a
functional assay

Sequencing Growth

<u>Cost of one human genome</u>

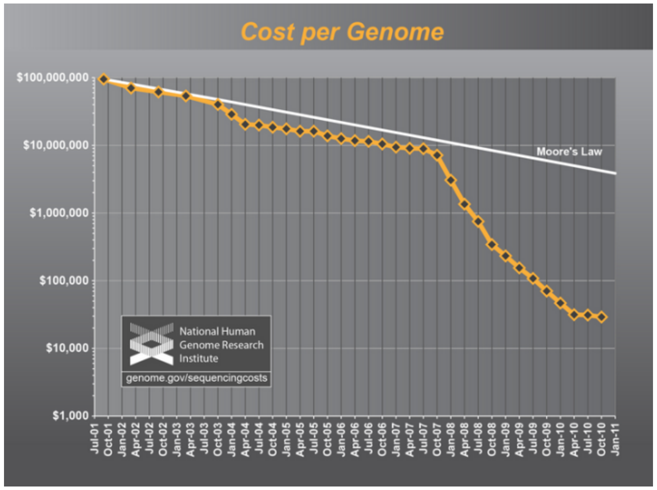
2004:	$30,000,000
2008:	$100,000
2010:	$10,000
2014:	**$1,000**
???:	$300



Baseline information

Cost of genome sequencing compared with Moore's law for computers

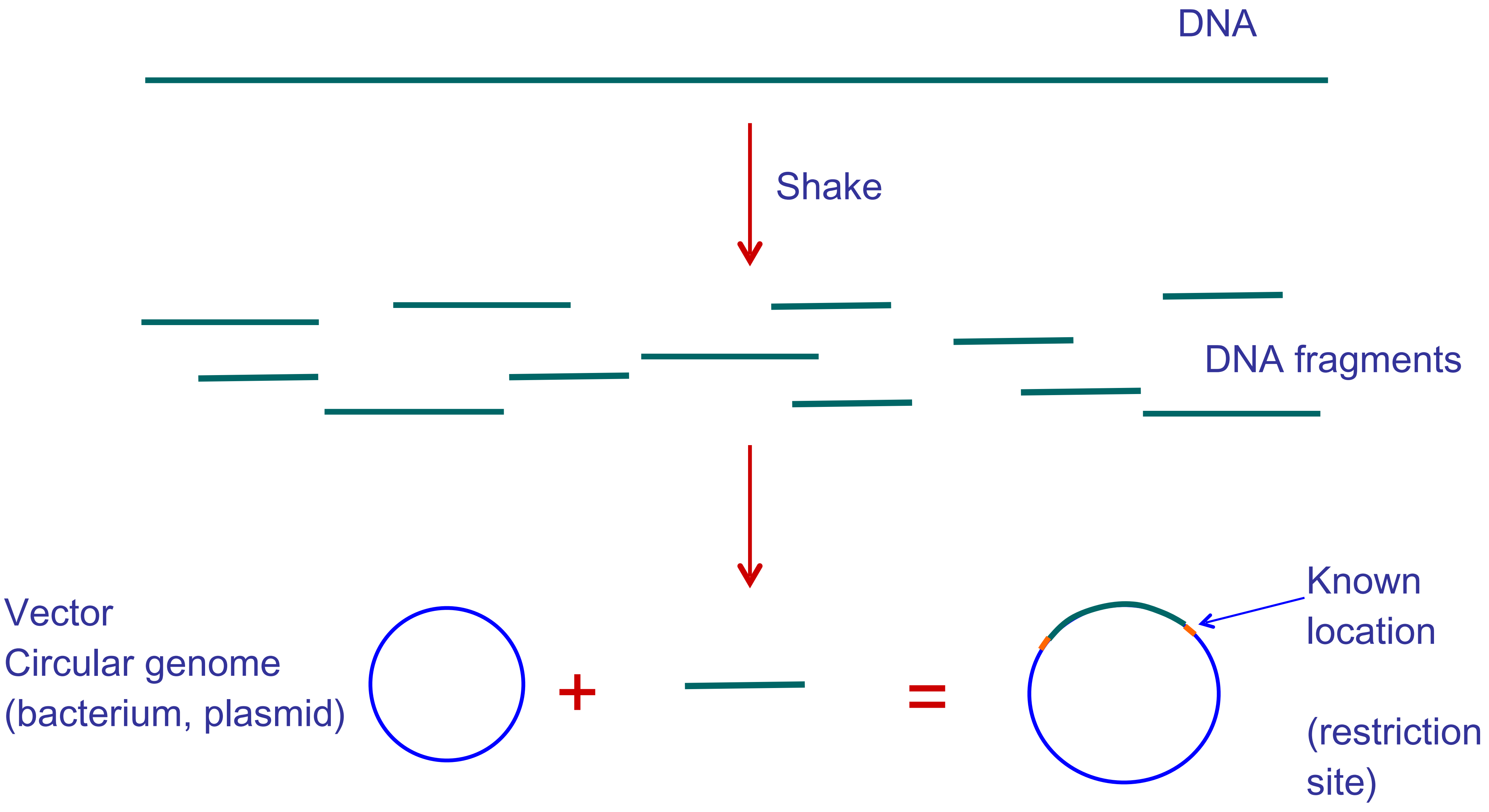
Source: Broad Institute



Cost per Genome

How much would you pay for a smartphone?

Ancient sequencing technology – Sanger Vectors



DNA

Shake

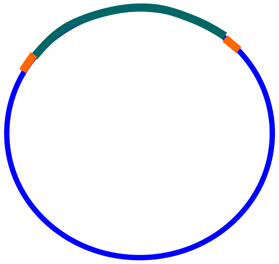
DNA fragments

Vector
Circular genome
(bacterium, plasmid)

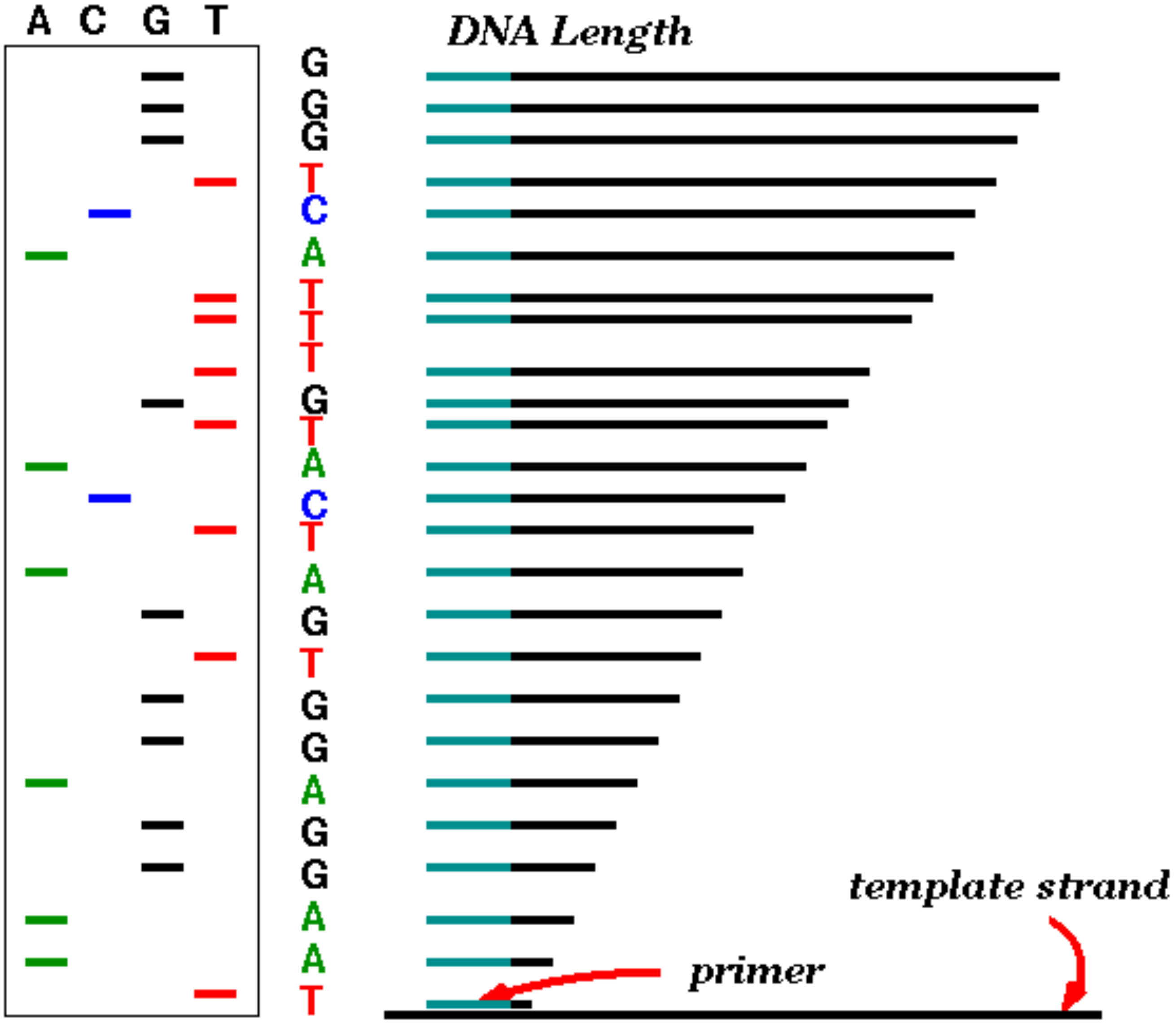
Known location

(restriction site)

Ancient sequencing technology – Sanger Gel Electrophoresis

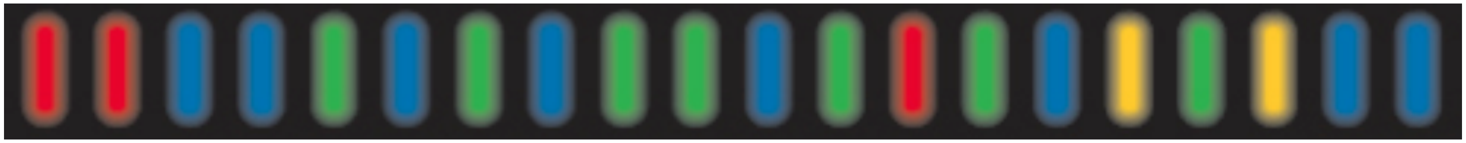


1. Start at primer (restriction site)
2. Grow DNA chain
3. Include dideoxynucleoside (modified a, c, g, t)
4. Stops reaction at all possible points
5. Separate products with length, using gel electrophoresis



Fluorescent Sanger sequencing trace

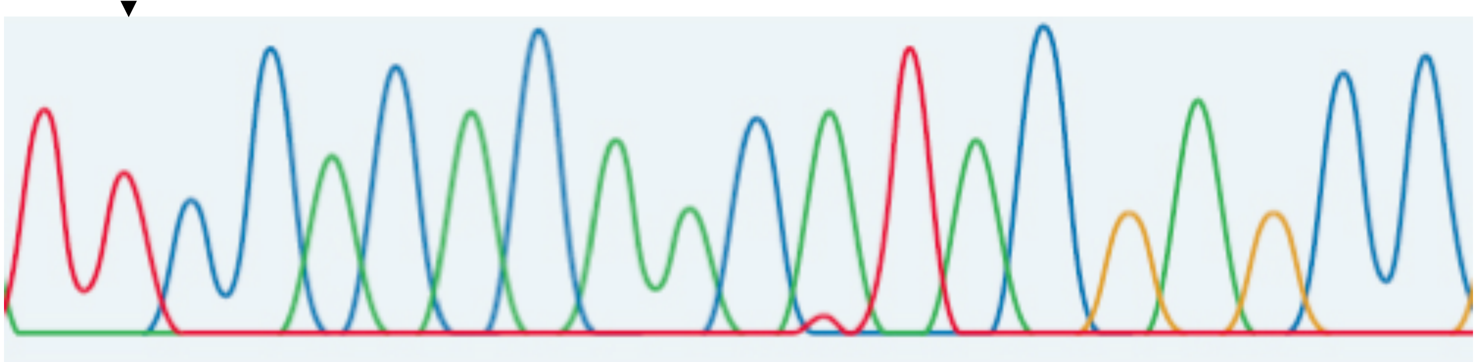
Lane signal



(Real fluorescent signals from a lane/capillary are much uglier than this).

A bunch of magic to boost signal/noise, correct for dye-effects, mobility differences, etc, generates the ‘final’ trace (for each capillary of the run)

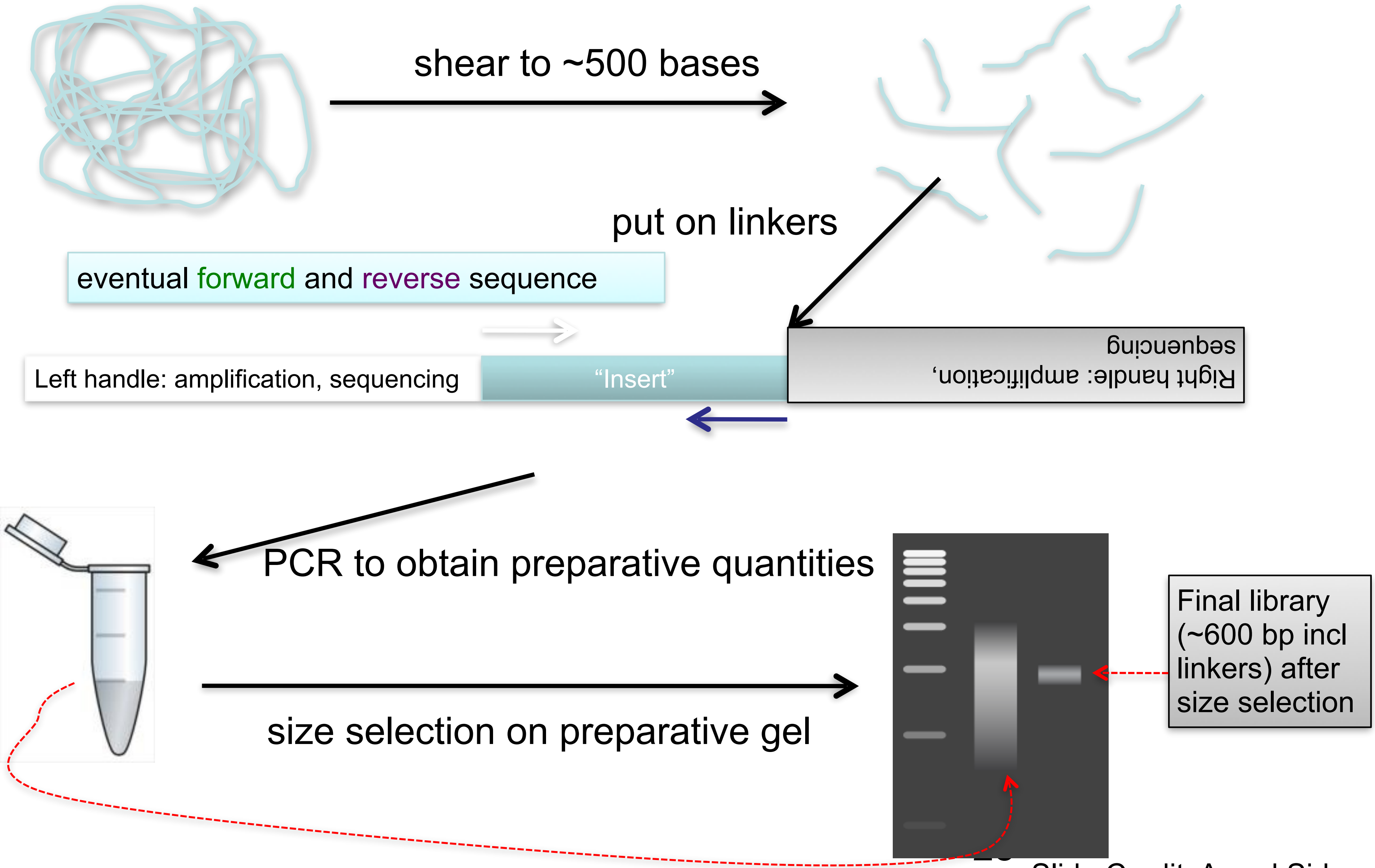
Trace



Recombinant DNA: Genes and Genomes. 3rd
Edition (Dec06). WH Freeman Press.

Slide Credit: Arend Sidow

Making a Library (present)



shear to ~500 bases

put on linkers

eventual forward and reverse sequence

Left handle: amplification, sequencing

“Insert”

Right handle: amplification, sequencing

PCR to obtain preparative quantities

size selection on preparative gel

Final library (~600 bp incl linkers) after size selection

Slide Credit: Arend Sidow

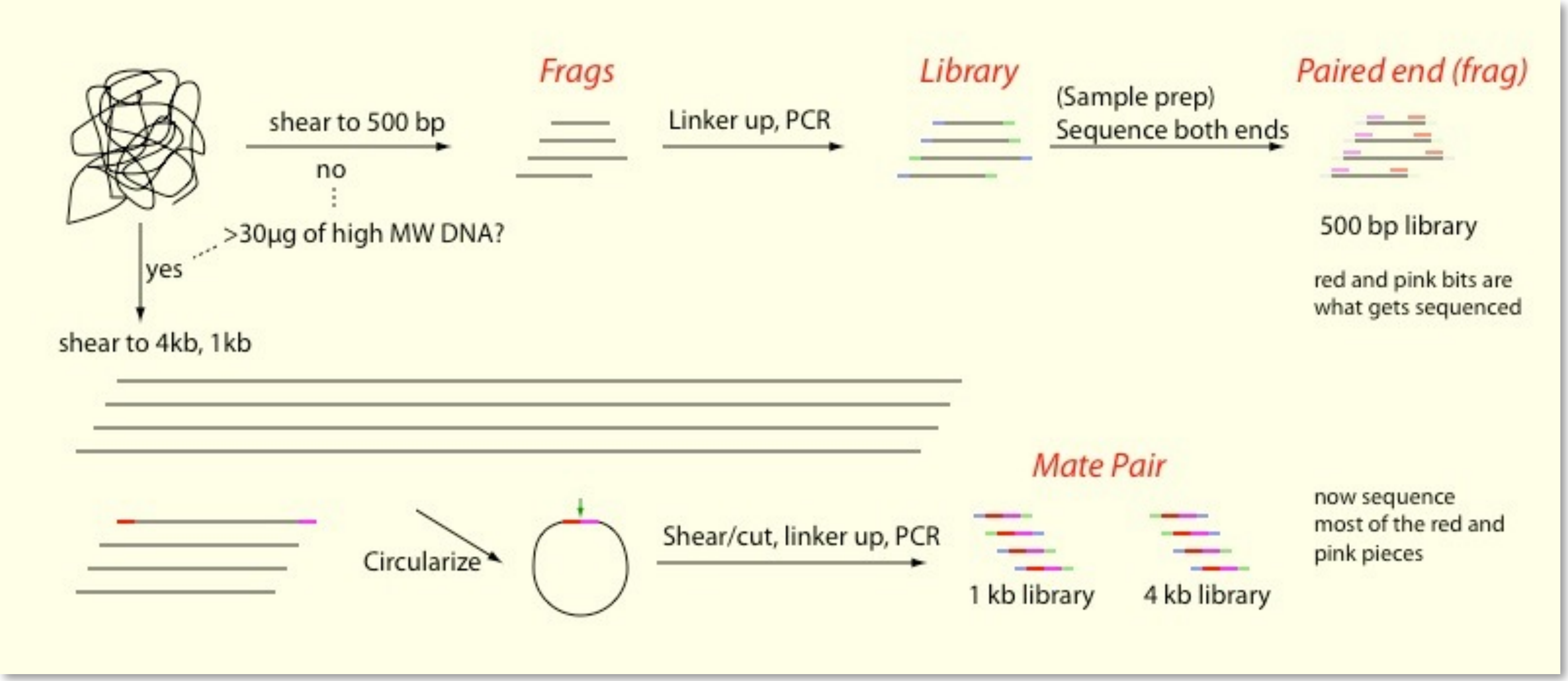
Library

- Library is a massively complex mix of -initially- individual, unique fragments

- Library amplification mildly amplifies each fragment to retain the complexity of the mix while obtaining preparative amounts
  - (how many-fold do 10 cycles of PCR amplify the sample?)

Slide Credit: Arend Sidow

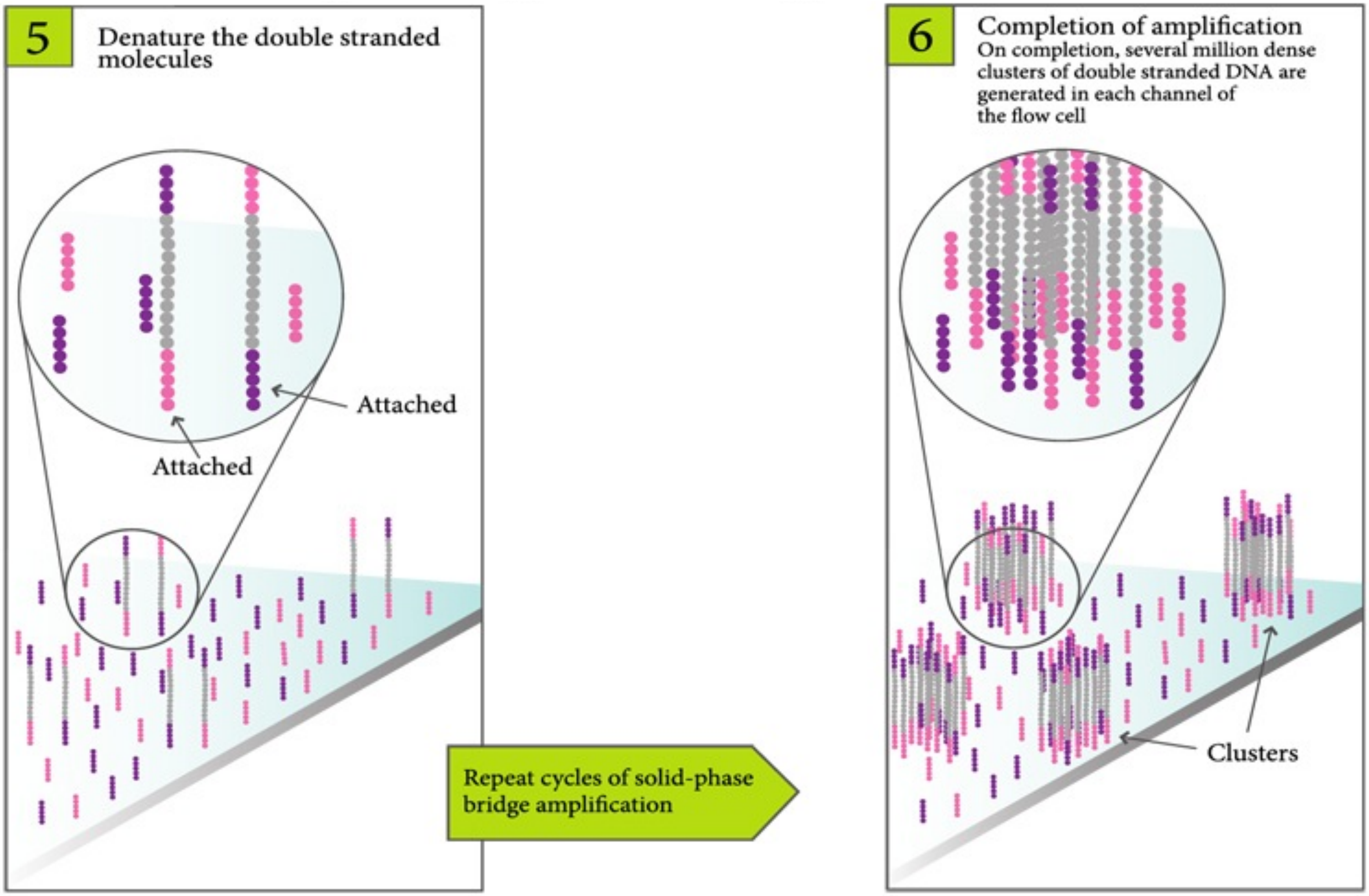
Fragment vs Mate pair (‘jumping’)



(Illumina has new kits/methods with which mate pair libraries can be built with less material)

Slide Credit: Arend Sidow

Illumina cluster concept



5 Denature the double stranded molecules

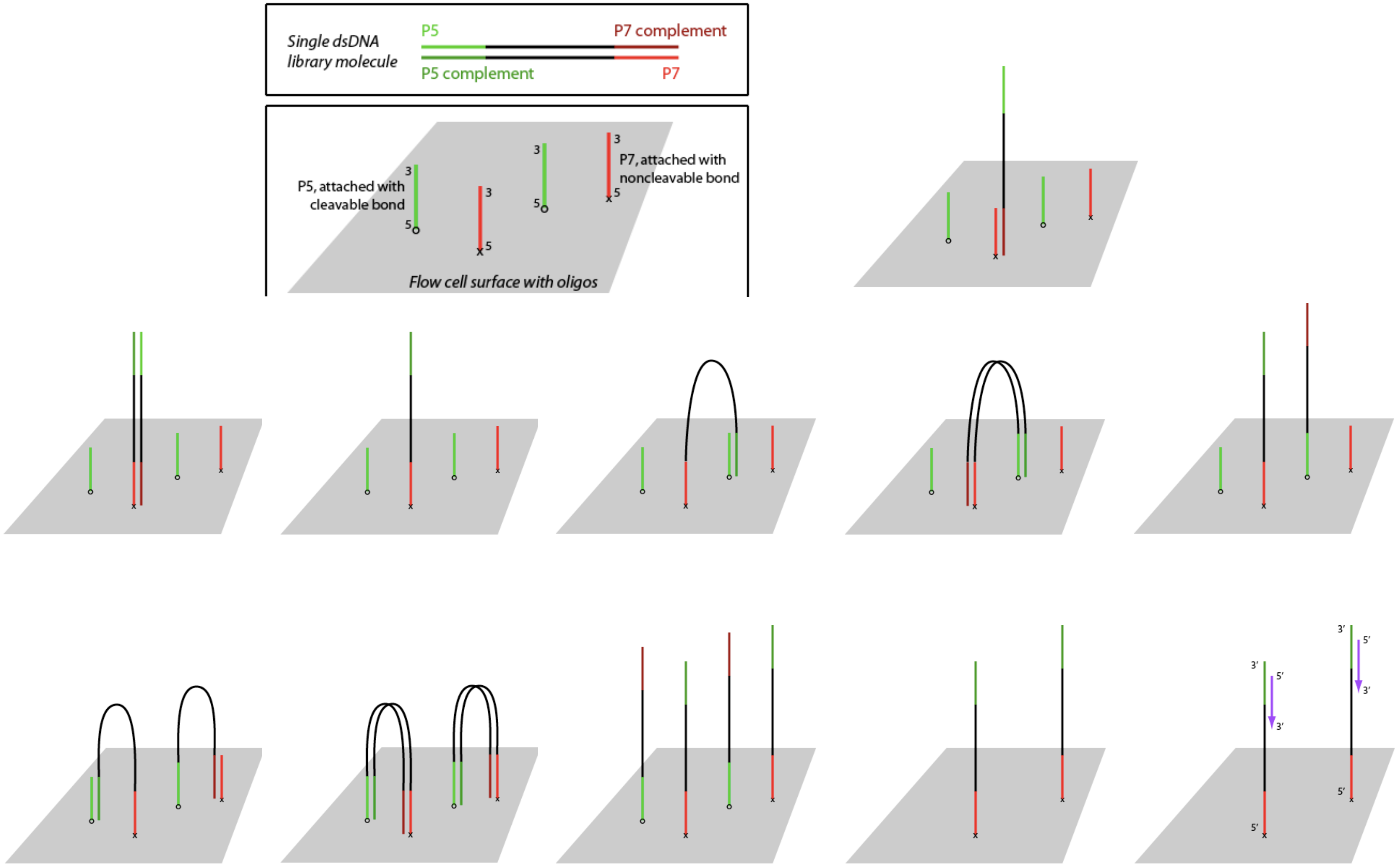
Repeat cycles of solid-phase bridge amplification

6 Completion of amplification

On completion, several million dense clusters of double stranded DNA are generated in each channel of the flow cell

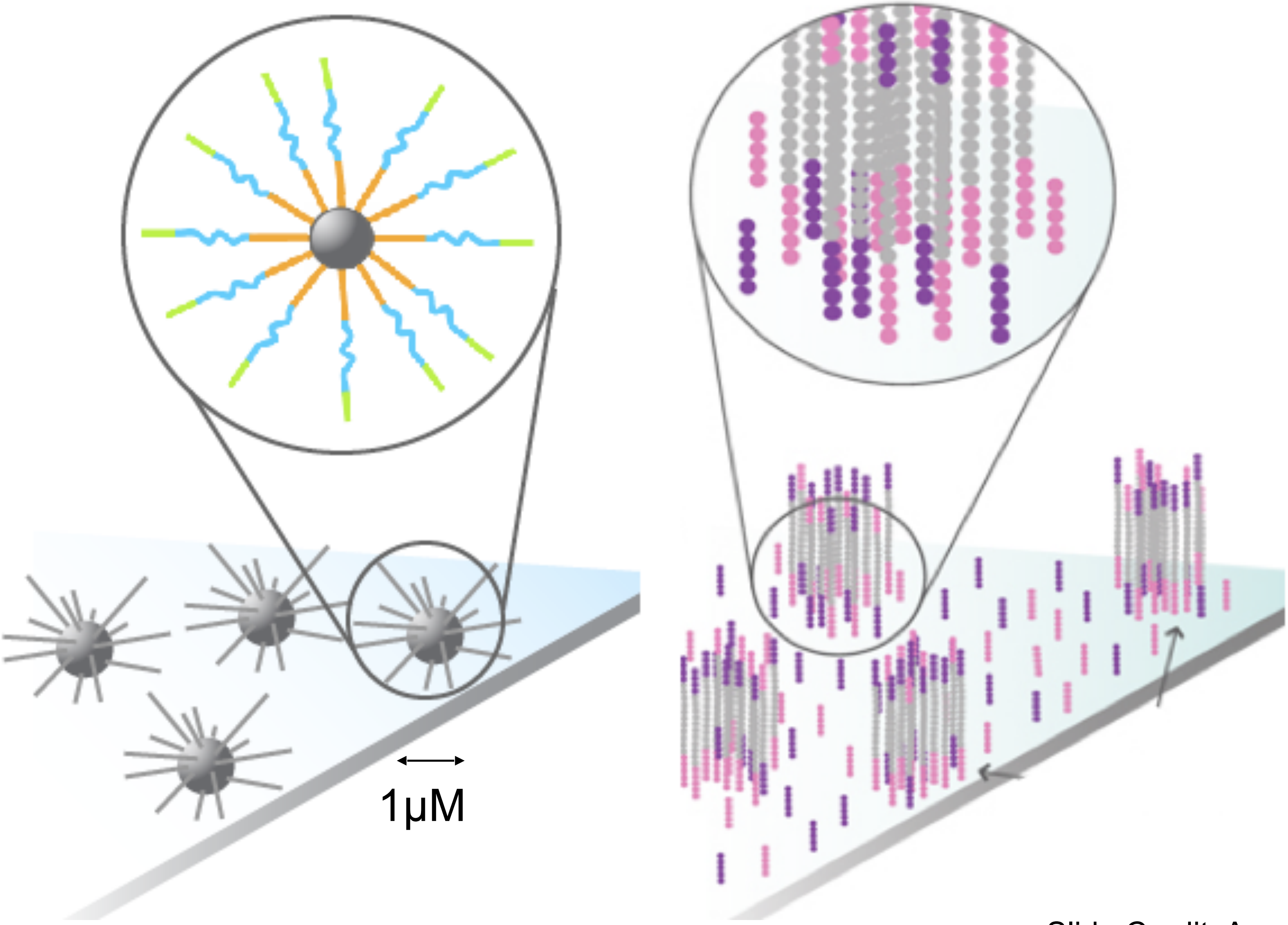
Slide Credit: Arend Sidow

Cluster generation (‘bridge amplification’)


Single dsDNA library molecule
P5
P7 complement
P5 complement
P7
P5, attached with cleavable bond
P7, attached with noncleavable bond
3
5
Flow cell surface with oligos
3′
5′

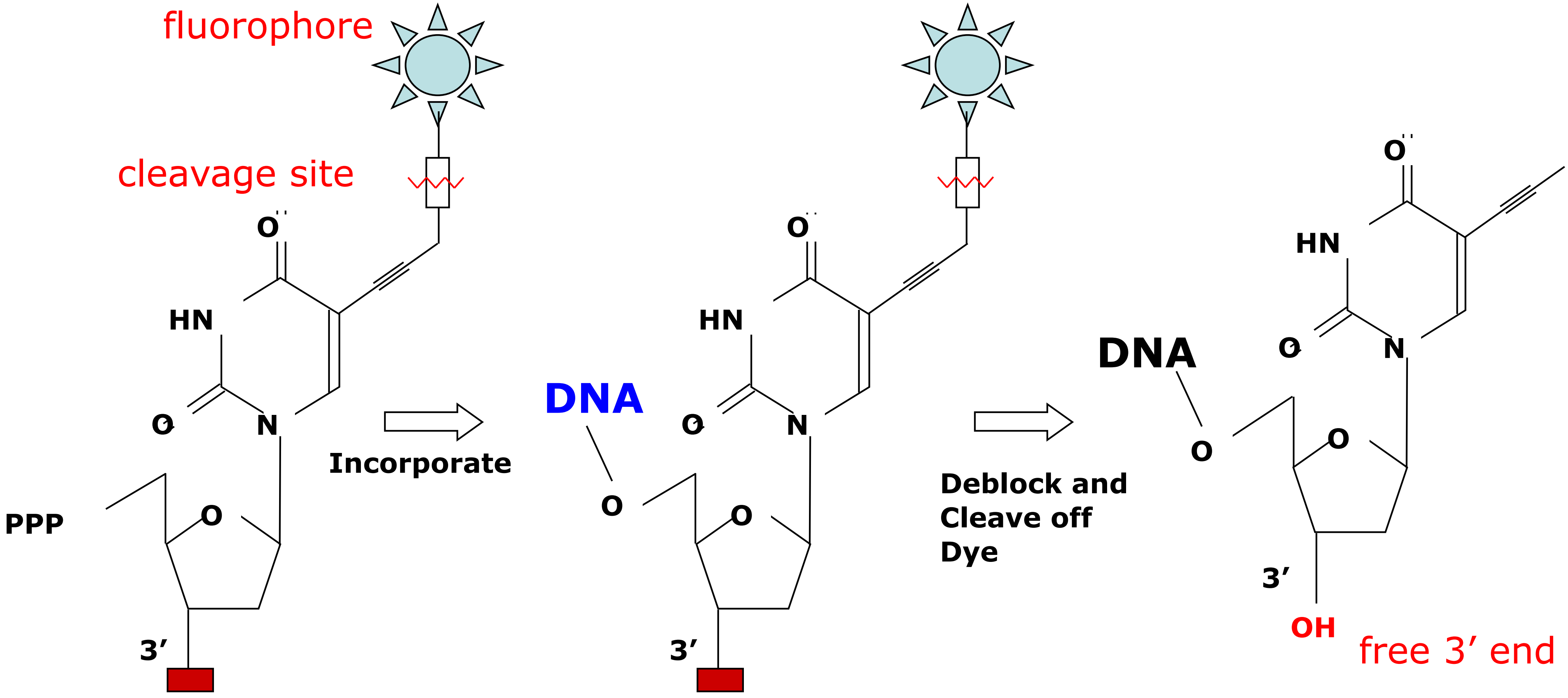
Slide Credit: Arend Sidow

Clonally Amplified Molecules on Flow Cell



Slide Credit: Arend Sidow

Reversible Terminators



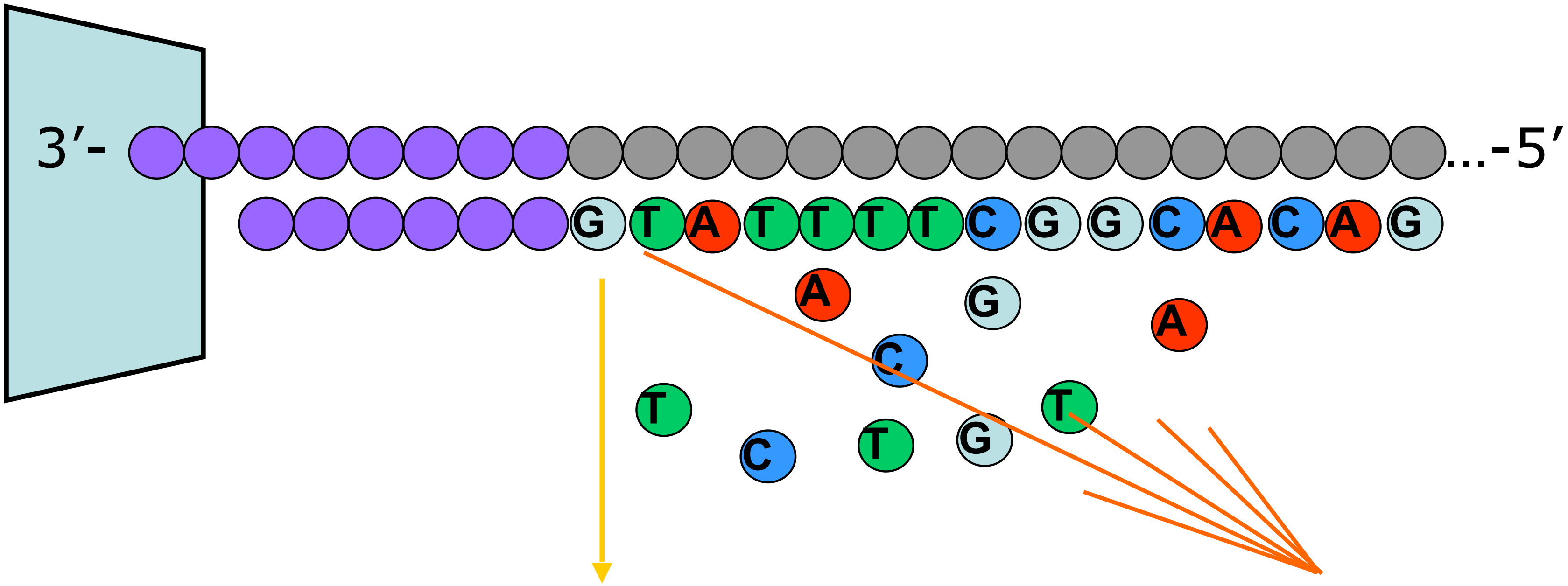
3' OH is blocked

Detection

Ready for Next Cycle

Slide Credit: Arend Sidow

Sequencing by Synthesis, One Base at a Time



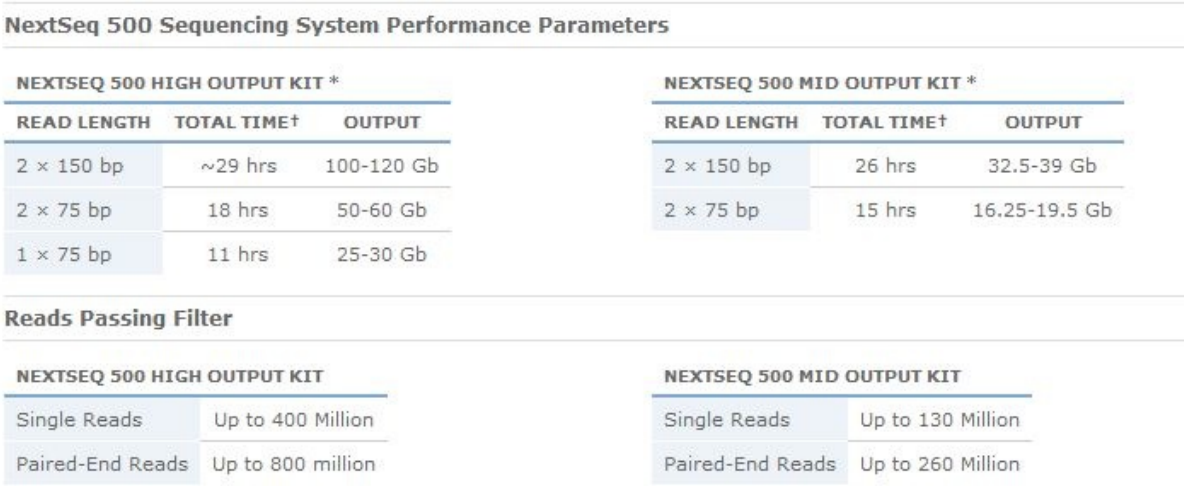
Cycle 1:	Add sequencing reagents
	First base incorporated
	Remove unincorporated bases
	Detect signal
Cycle 2-n:	Add sequencing reagents and repeat

Solexa

Slide Credit: Arend Sidow

HiSeq X & NextSeq

Preliminary specs:
Run time: 3 days
Output: 1.6 Tb
#reads: $6x10^9$
Read length: 2x150bp



NextSeq 500 Sequencing System Performance Parameters

NEXTSEQ 500 HIGH OUTPUT KIT *

READ LENGTH	TOTAL TIME†	OUTPUT
2 × 150 bp	~29 hrs	100-120 Gb
2 × 75 bp	18 hrs	50-60 Gb
1 × 75 bp	11 hrs	25-30 Gb

NEXTSEQ 500 MID OUTPUT KIT *

READ LENGTH	TOTAL TIME†	OUTPUT
2 × 150 bp	26 hrs	32.5-39 Gb
2 × 75 bp	15 hrs	16.25-19.5 Gb

Reads Passing Filter

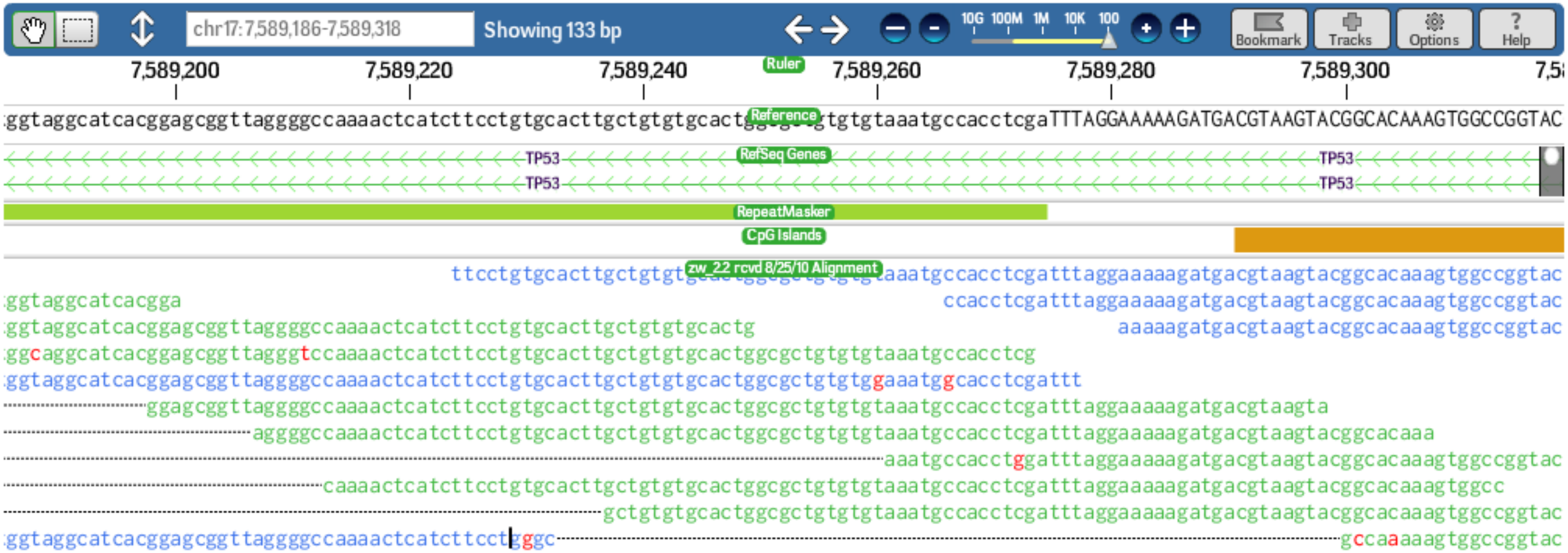
NEXTSEQ 500 HIGH OUTPUT KIT

Single Reads	Up to 400 Million
Paired-End Reads	Up to 800 million

NEXTSEQ 500 MID OUTPUT KIT

Single Reads	Up to 130 Million
Paired-End Reads	Up to 260 Million

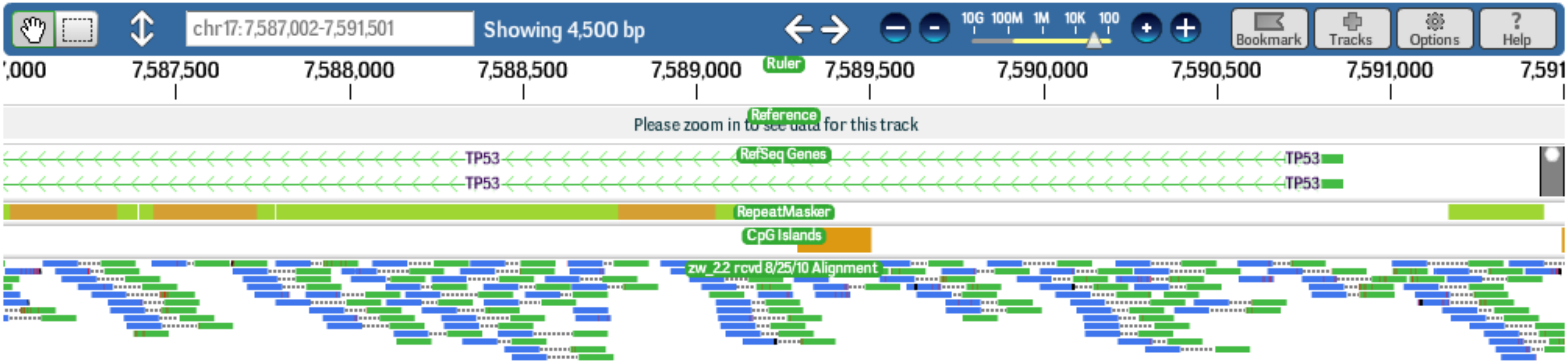
Read Mapping



chr17:7,589,186-7,589,318 Showing 133 bp

```
ggtaggcatcacggagcggttaggggccaaaactcatcttcctgtgcacttgctgtgtgcactg      tgtgtaaatgccacctcgaTTTAGGAAAAAGATGACGTAAGTACGGCACAAAGTGGCCGGTAC

                                      ttcctgtgcacttgctgtgt                 aaatgccacctcgatttaggaaaaagatgacgtaagtacggcacaaagtggccggtac
ggtaggcatcacgga                                                                 ccacctcgatttaggaaaaagatgacgtaagtacggcacaaagtggccggtac
ggtaggcatcacggagcggttaggggccaaaactcatcttcctgtgcacttgctgtgtgcactg                               aaaaagatgacgtaagtacggcacaaagtggccggtac
ggcaggcatcacggagcggttagggtccaaaactcatcttcctgtgcacttgctgtgtgcactggcgctgtgtgtaaatgccacctcg
ggtaggcatcacggagcggttaggggccaaaactcatcttcctgtgcacttgctgtgtgcactggcgctgtgtggaaatggcacctcgattt
............ggagcggttaggggccaaaactcatcttcctgtgcacttgctgtgtgcactggcgctgtgtgtaaatgccacctcgatttaggaaaaagatgacgtaagta
.....................aggggccaaaactcatcttcctgtgcacttgctgtgtgcactggcgctgtgtgtaaatgccacctcgatttaggaaaaagatgacgtaagtacggcacaaa
...........................................................................aaatgccacctggatttaggaaaaagatgacgtaagtacggcacaaagtggccggtac
...........................caaaactcatcttcctgtgcacttgctgtgtgcactggcgctgtgtgtaaatgccacctcgatttaggaaaaagatgacgtaagtacggcacaaagtggcc
...................................................gctgtgtgcactggcgctgtgtgtaaatgccacctcgatttaggaaaaagatgacgtaagtacggcacaaagtggccggtac
ggtaggcatcacggagcggttaggggccaaaactcatcttcct|ggc....................................................................gccaaaagtggccggtac
```

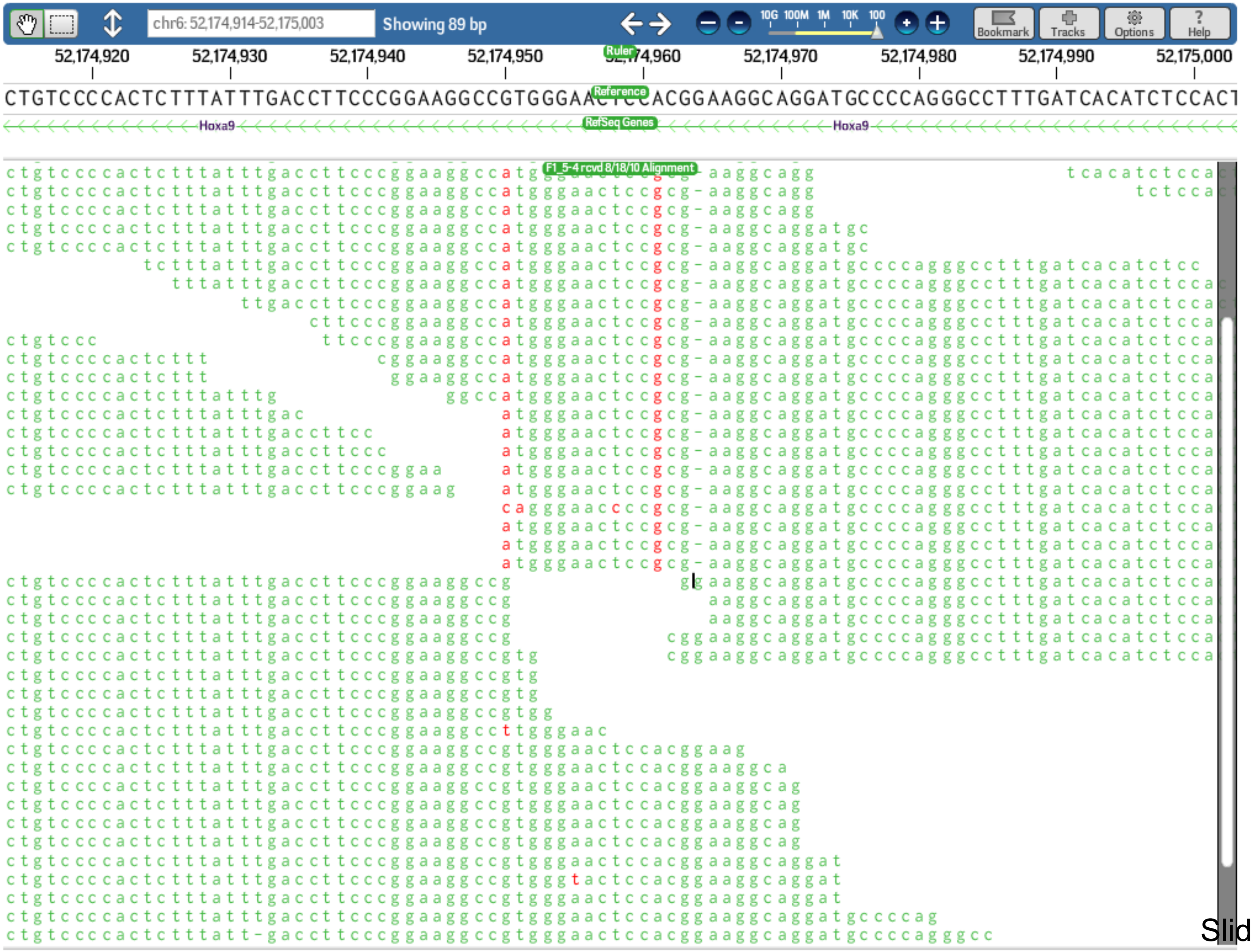


chr17:7,587,002-7,591,501 Showing 4,500 bp

Please zoom in to see data for this track

Slide Credit: Arend Sidow

Variation Discovery



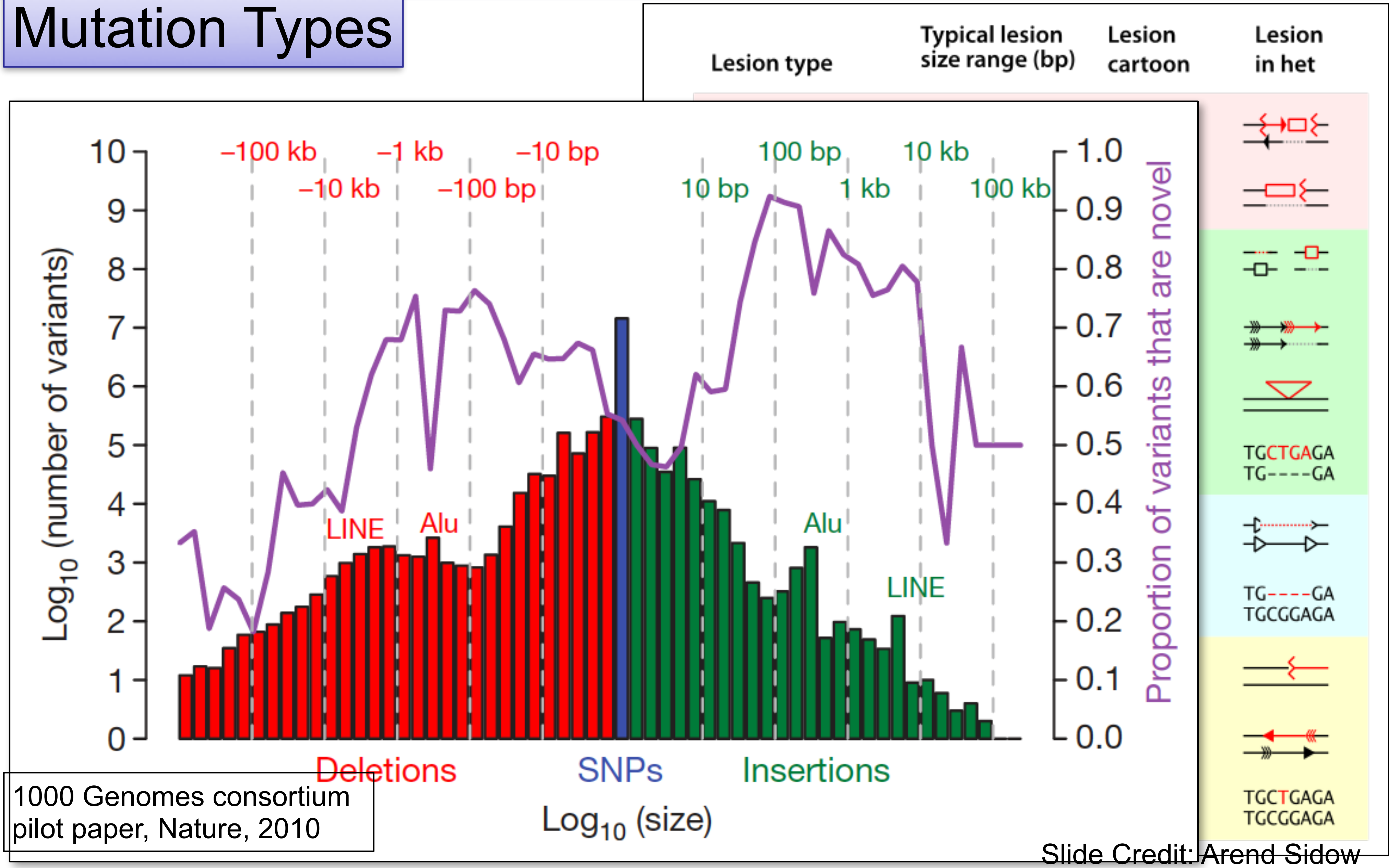
```
CTGTCCCACTCTTTATTTGACCTTCCCGGAAGGCCGTGGGAACTCCACGGAAGGCAGGATGCCCCAGGGCCTTTGATCACATCTCCACT

ctgtcccactctttatttgaccttcccggaaggccatg           -aaggcagg                  tcacatctcca
ctgtcccactctttatttgaccttcccggaaggccatgggaactccgcg-aaggcagg                       tctcca
ctgtcccactctttatttgaccttcccggaaggccatgggaactccgcg-aaggcagg
ctgtcccactctttatttgaccttcccggaaggccatgggaactccgcg-aaggcaggatgc
ctgtcccactctttatttgaccttcccggaaggccatgggaactccgcg-aaggcaggatgc
         tctttatttgaccttcccggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcc
           tttatttgaccttcccggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                ttgaccttcccggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                     cttcccggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtccc               ttcccggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactcttt            cggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactcttt             ggaaggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttg            ggccatgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgac              atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcc         atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttccc        atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaa    atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaag   atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                                   cagggaacccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                                   atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                                   atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
                                   atgggaactccgcg-aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccg            ggaaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccg              aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccg              aaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccg           cggaaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccgtg         cggaaggcaggatgccccagggcctttgatcacatctcca
ctgtcccactctttatttgaccttcccggaaggccgtg
ctgtcccactctttatttgaccttcccggaaggccgtg
ctgtcccactctttatttgaccttcccggaaggccgtgg
ctgtcccactctttatttgaccttcccggaaggccttgggaac
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaag
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggca
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcag
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcag
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcag
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcag
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcaggat
ctgtcccactctttatttgaccttcccggaaggccgtgggtactccacggaaggcaggat
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcaggat
ctgtcccactctttatttgaccttcccggaaggccgtgggaactccacggaaggcaggatgccccag
ctgtcccactctttatt-gaccttcccggaaggccgtgggaactccacggaaggcaggatgccccagggcc
```

Slide Credit: Arend Sidow

Amount of variation – types of lesions

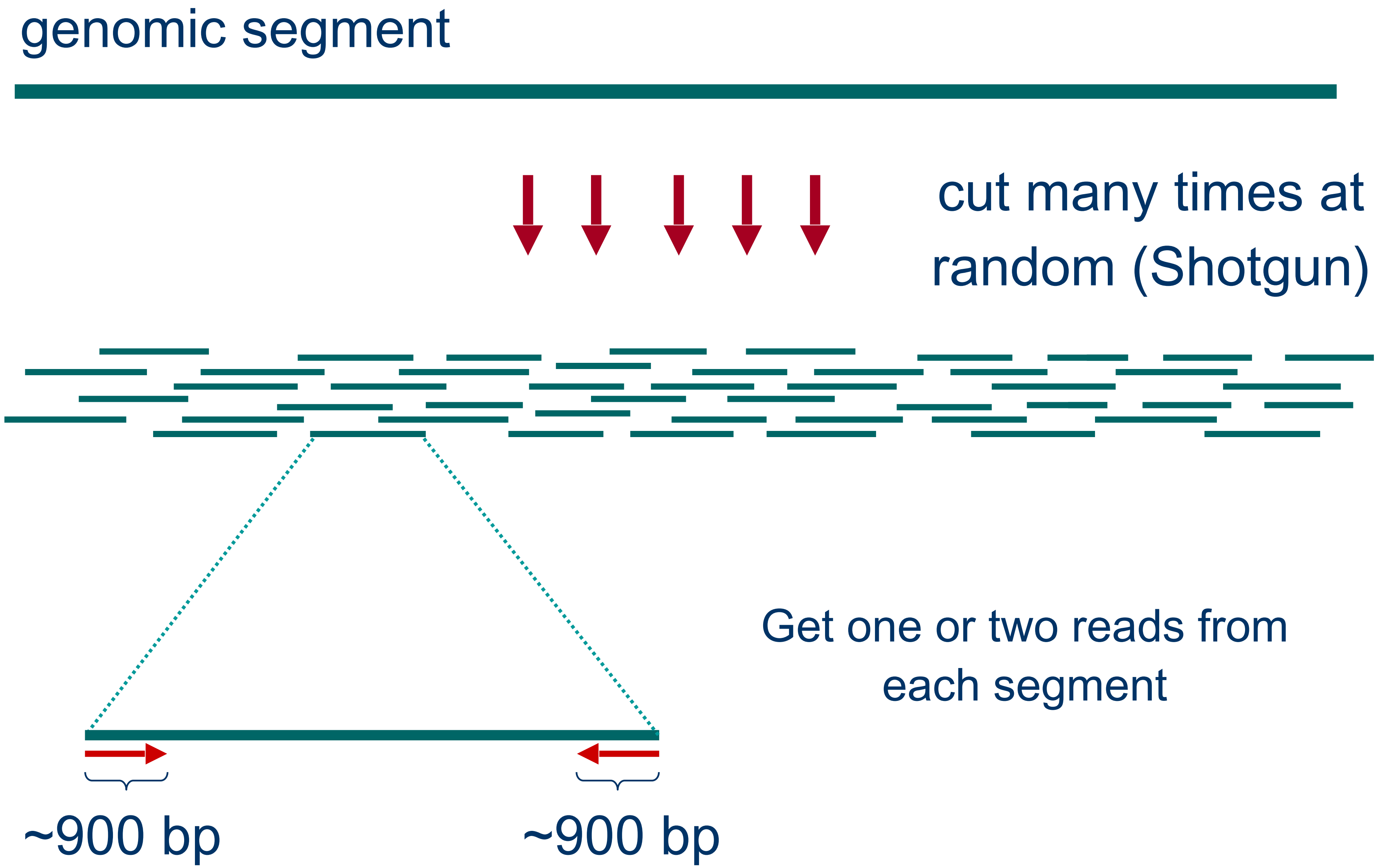
Mutation Types



1000 Genomes consortium pilot paper, Nature, 2010

Slide Credit: Arend Sidow

Method to sequence longer regions



genomic segment

cut many times at random (Shotgun)

Get one or two reads from each segment

~900 bp

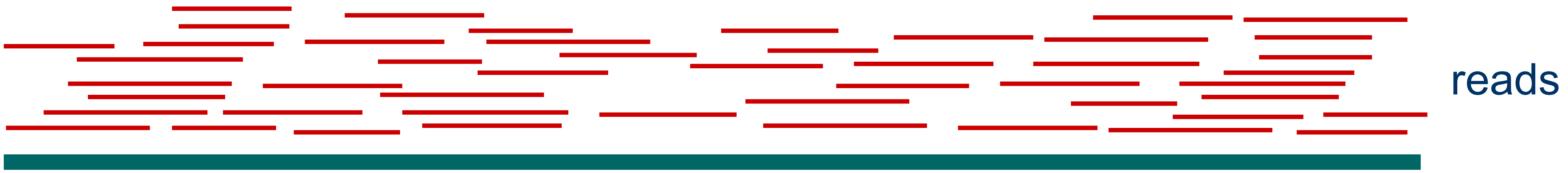
~900 bp

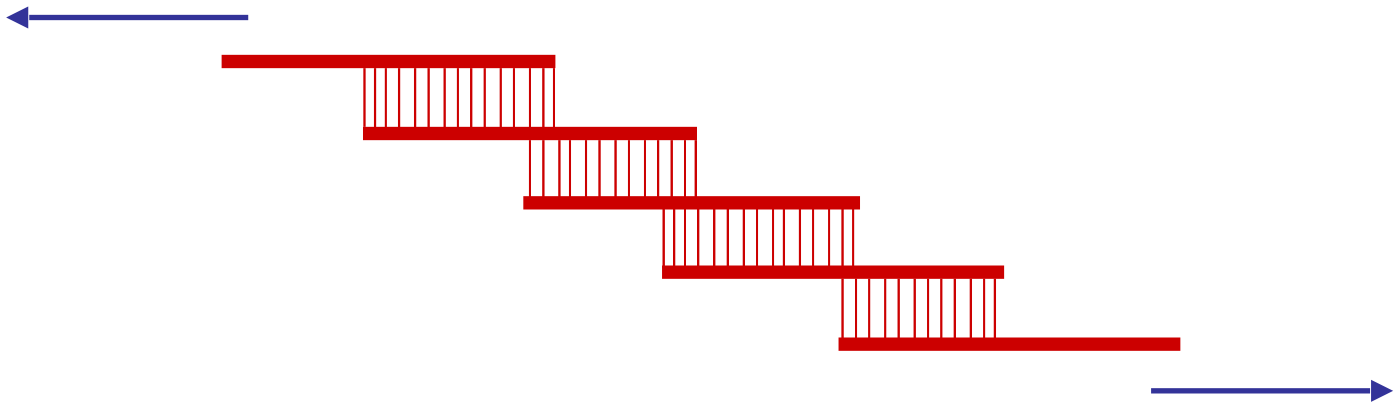
Two main assembly problems

- De Novo Assembly

- Resequencing

Reconstructing the Sequence (De Novo Assembly)

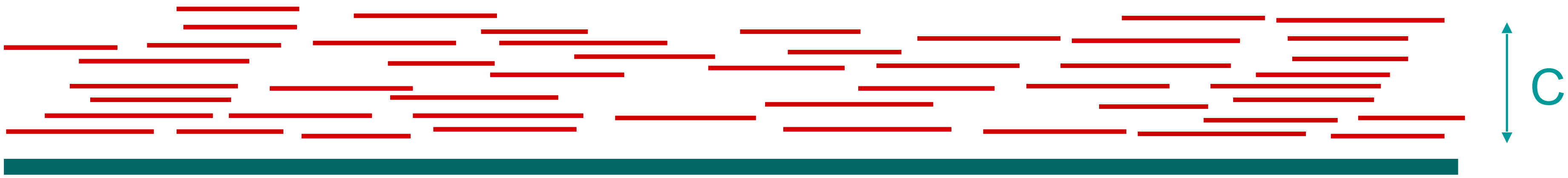




Cover region with high redundancy

Overlap & extend reads to reconstruct the original genomic region

Definition of Coverage



Length of genomic segment: **G**
Number of reads: **N**
Length of each read: **L**

Definition: Coverage **C = N L / G**

How much coverage is enough?

Lander-Waterman model: Prob[not covered bp] = e^{-C}
Assuming uniform distribution of reads, C=10 results in 1 gapped region /1,000,000 nucleotides

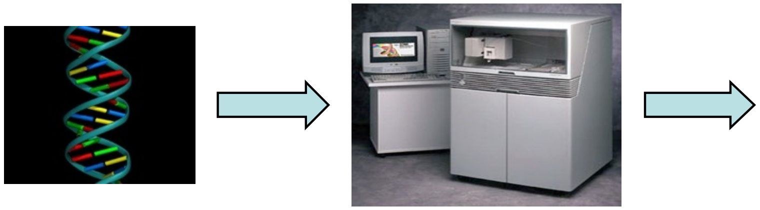
Repeats

Bacterial genomes: 5%
Mammals: 50%

Repeat types:

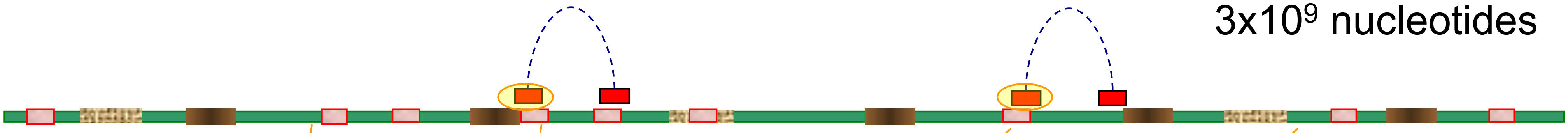
- **Low-Complexity DNA** (e.g. ATATATATACATA…)
- **Microsatellite repeats** $(a_1 \ldots a_k)^N$ where k ~ 3-6
  (e.g. CAGCAGTAGCAGCACCAG)
- **Transposons**
  - SINE (Short Interspersed Nuclear Elements)
    e.g., ALU: ~300-long, 10^6 copies
  - LINE (Long Interspersed Nuclear Elements)
    ~4000-long, 200,000 copies
  - LTR retroposons (Long Terminal Repeats (~700 bp) at each end)
    cousins of HIV
- **Gene Families** genes duplicate & then diverge (paralogs)
- **Recent duplications** ~100,000-long, very similar copies

Sequencing and Fragment Assembly

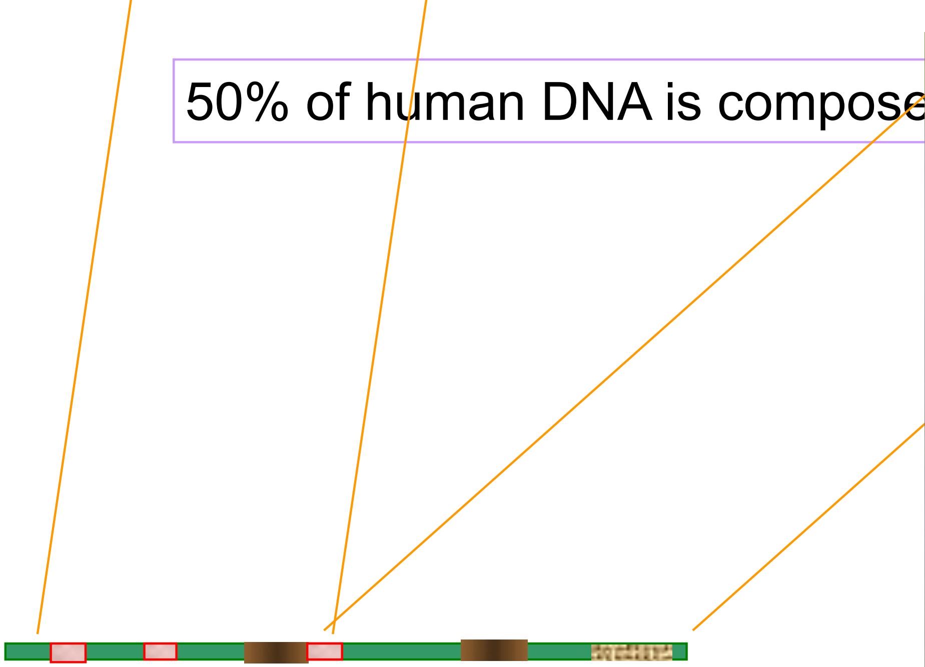


```
AGTAGCACAGAC
TACGACGAGACG
ATCGTGCGAGCG
ACGGCGTAGTGT
GCTGTACTGTCG
TGTGTGTGTACT
CTCCT
```

3×10^9 nucleotides



50% of human DNA is compose

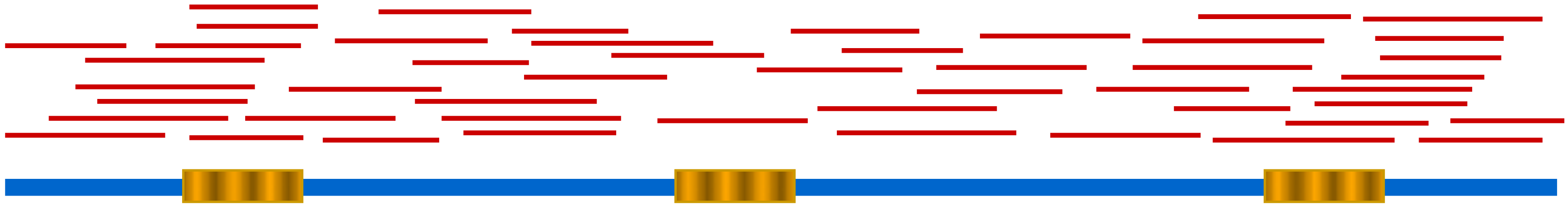


Error!
Glued together two distant regions

What can we do about repeats?

Two main approaches:

- Cluster the reads

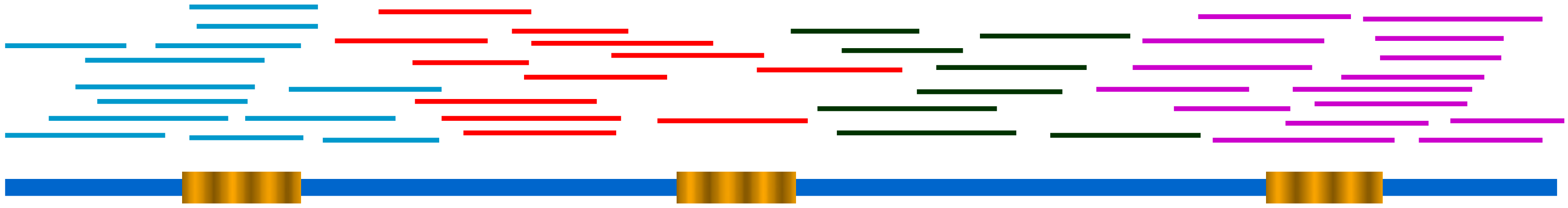


- Link the reads

What can we do about repeats?

Two main approaches:

- Cluster the reads

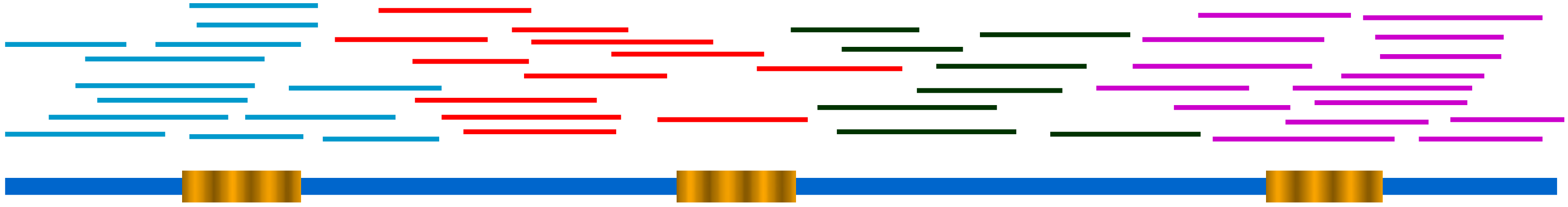


- Link the reads

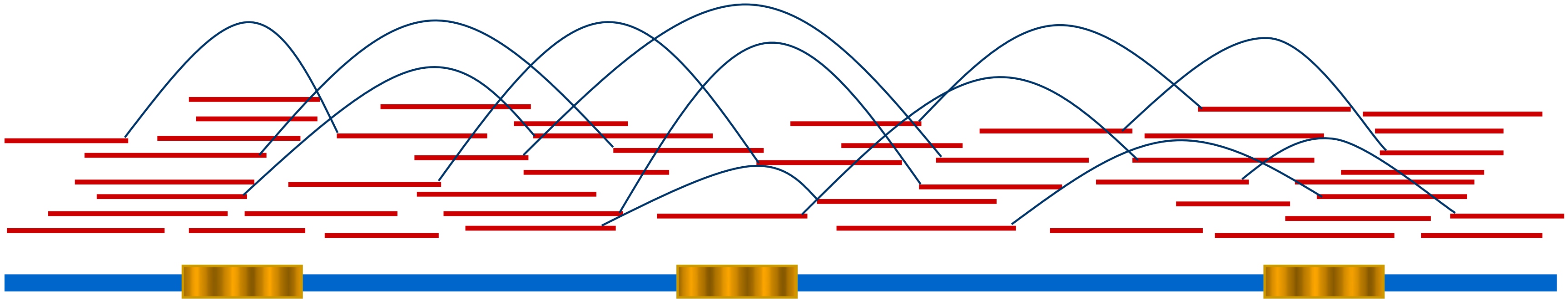
What can we do about repeats?

Two main approaches:

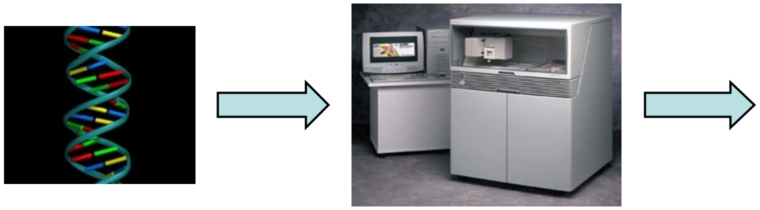
- Cluster the reads



- Link the reads

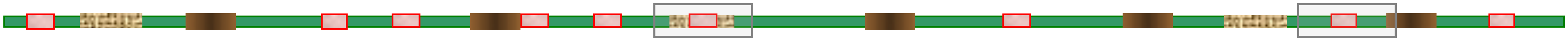


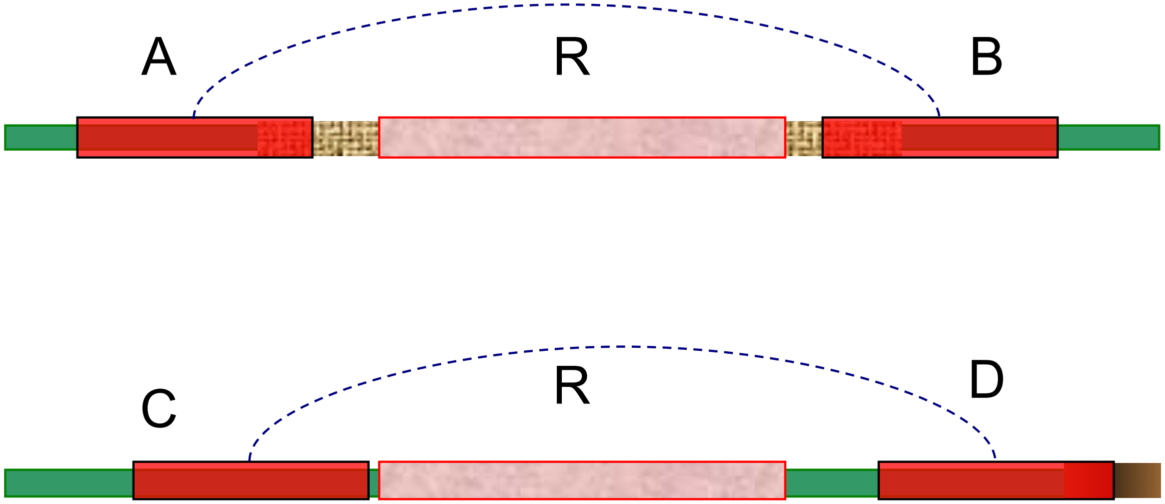
Sequencing and Fragment Assembly



```
AGTAGCACAGAC
TACGACGAGACG
ATCGTGCGAGCG
ACGGCGTAGTGT
GCTGTACTGTCG
TGTGTGTGTACT
CTCCT
```

$3x10^9$ nucleotides



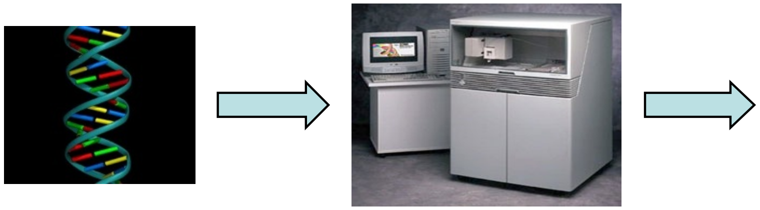


ARB, CRD

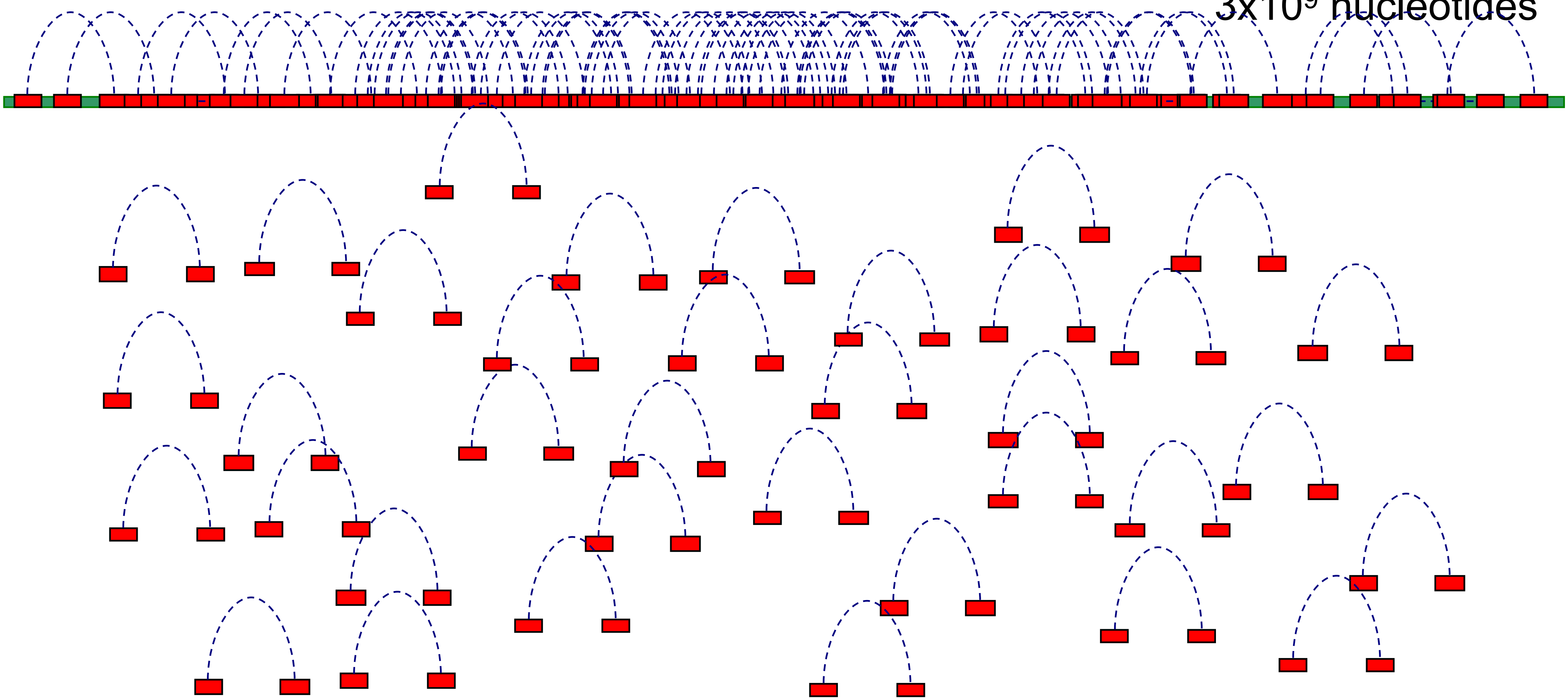
or

~~ARD, CRB~~ ?

Sequencing and Fragment Assembly



```
AGTAGCACAGAC
TACGACGAGACG
ATCGTGCGAGCG
ACGGCGTAGTGT
GCTGTACTGTCG
TGTGTGTGTACT
CTCCT
```

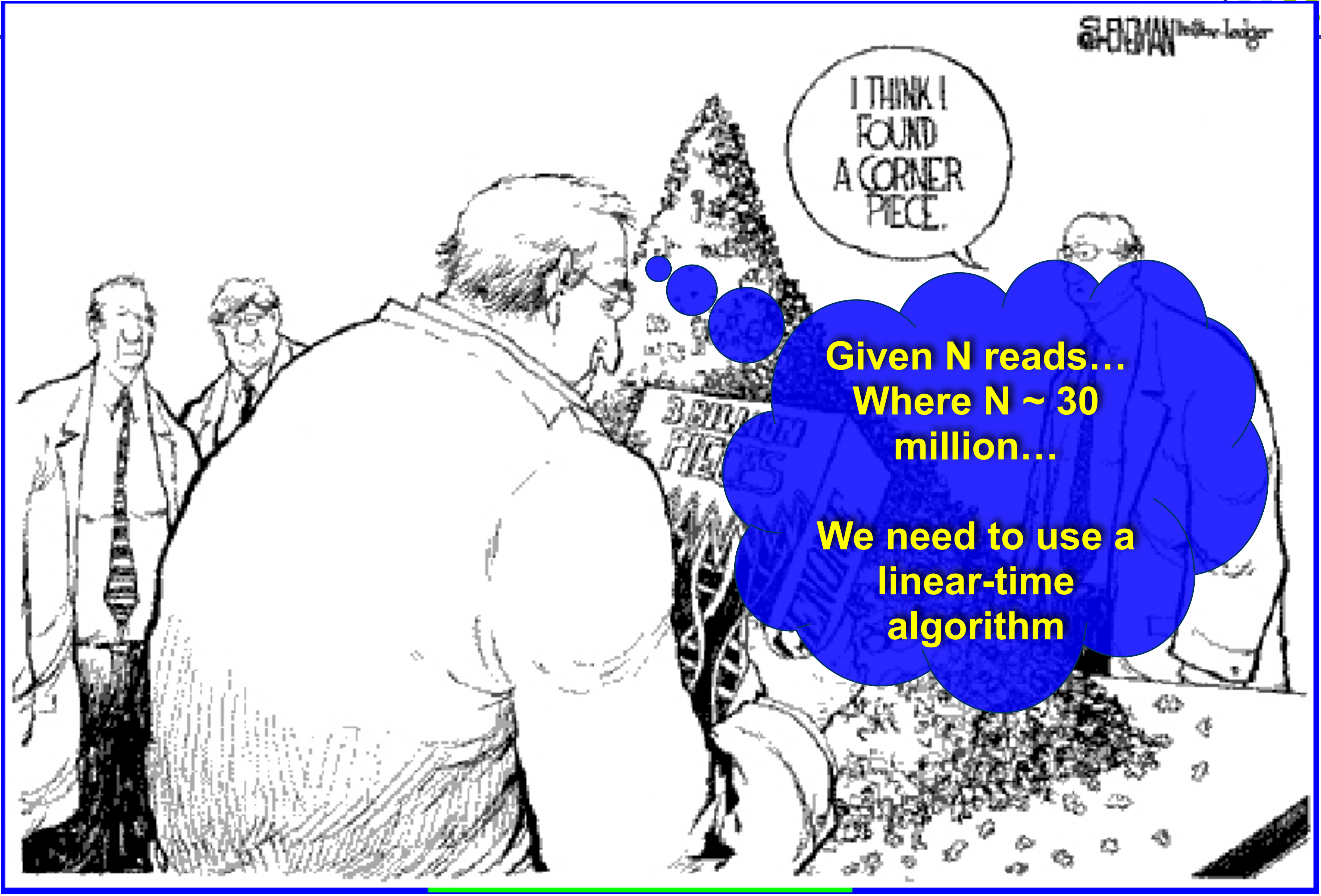


3x10^9 nucleotides

Fragment Assembly

(in whole-genome shotgun sequencing)

Fragment Assembly



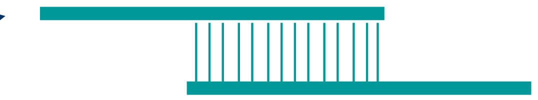
Given N reads…
Where N ~ 30 million…

We need to use a linear-time algorithm

Steps to Assemble a Genome

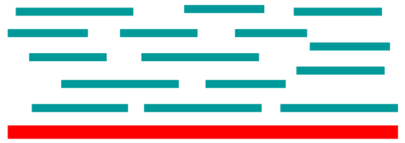
Some Terminology

read a 500-900 long word that comes out of sequencer

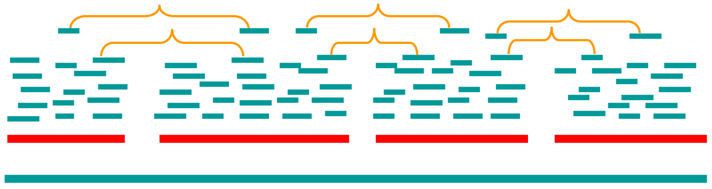


mate pair a pair of reads from two ends of the same insert fragment

contig a contiguous sequence formed by several overlapping reads with no gaps



supercontig (scaffold) an ordered and oriented set of contigs, usually by mate pairs



consensus sequene sequence derived from the multiple alignment of reads in a contig

..ACGATTACAATAGGTT..

1. Find Overlapping Reads

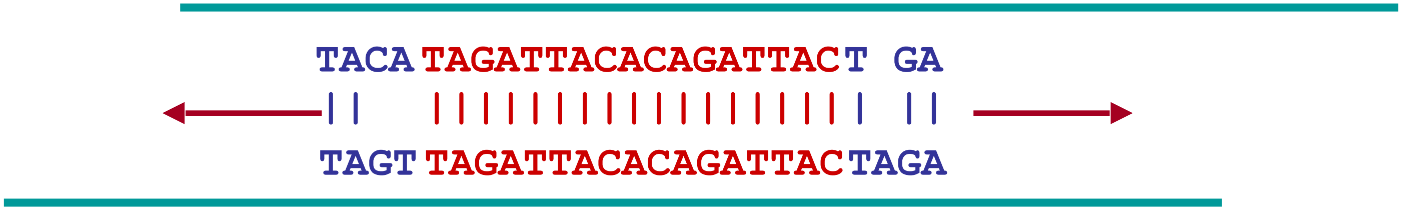
```
aaactgcagtacggatct
aaactgcag
 aactgcagt
…
        gtacggatct
         tacggatct
gggcccaaactgcagtac
gggcccaaa
 ggcccaaac
…
        actgcagta
         ctgcagtac
gtacggatctactacaca
gtacggatc
 tacggatct
…
        ctactacac
         tactacaca
```

```
(read, pos., word, orient.)
aaactgcag
aactgcagt
actgcagta
…
gtacggatc
tacggatct
gggcccaaa
ggcccaaac
gcccaaact
…
actgcagta
ctgcagtac
gtacggatc
tacggatct
acggatcta
…
ctactacac
tactacaca
```

```
(word, read, orient., pos.)
aaactgcag
aactgcagt
acggatcta
actgcagta
actgcagta
cccaaactg
cggatctac
ctactacac
ctgcagtac
ctgcagtac
gcccaaact
ggcccaaac
gggcccaaa
gtacggatc
gtacggatc
tacggatct
tacggatct
tactacaca
```

1. Find Overlapping Reads

- Find pairs of reads sharing a k-mer, k ~ 24
- Extend to full alignment – throw away if not >98% similar



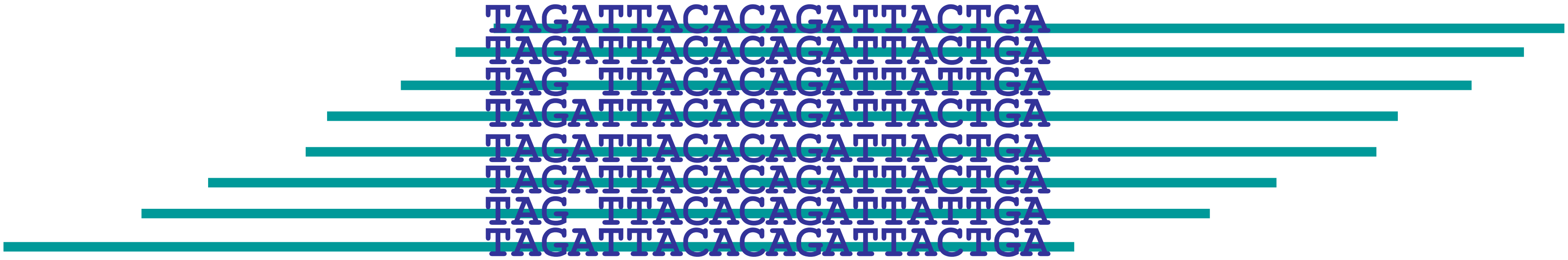
TACA TAGATTACACAGATTACT GA

TAGT TAGATTACACAGATTAC TAGA

- Caveat: repeats
  - A k-mer that occurs N times, causes O(N^2) read/read comparisons
  - ALU k-mers could cause up to 1,000,000^2 comparisons
- Solution:
  - Discard all k-mers that occur “too often”
    - Set cutoff to balance sensitivity/speed tradeoff, according to genome at hand and computing resources available

1. Find Overlapping Reads

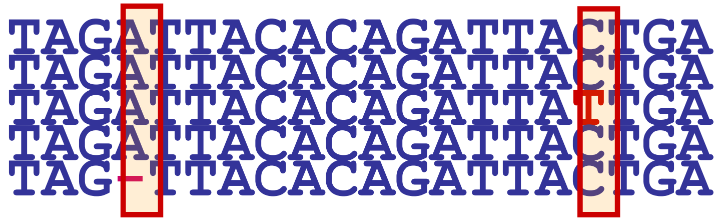
Create local multiple alignments from the overlapping reads



```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
```

1. Find Overlapping Reads

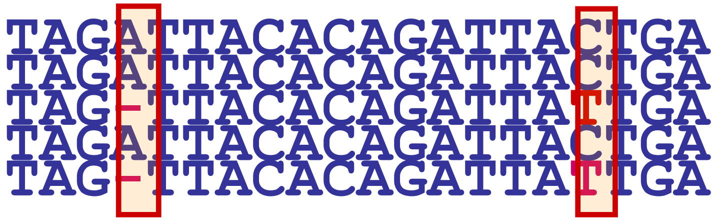
- Correct errors using multiple alignment



```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTATTGA
TAGATTACACAGATTACTGA
TAG-TTACACAGATTACTGA
```

insert A

replace T with C



```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG-TTACACAGATTATTGA
TAGATTACACAGATTACTGA
TAG-TTACACAGATTATTGA
```

correlated errors—
probably caused by repeats
⇒ disentangle overlaps

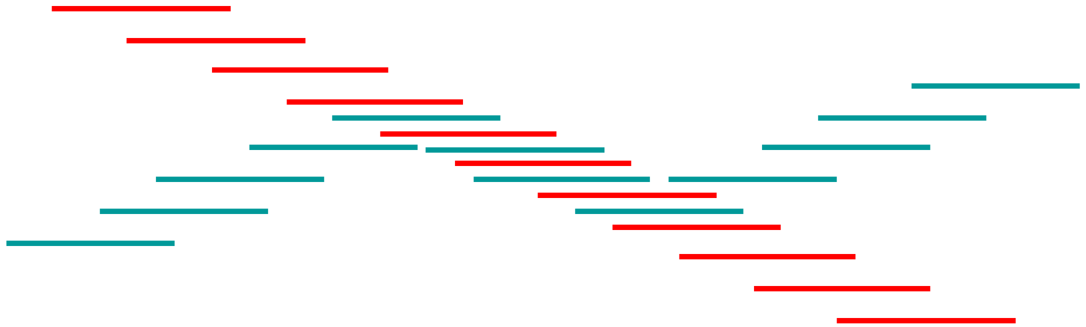
```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
```

```
TAG-TTACACAGATTATTGA
TAG-TTACACAGATTATTGA
```

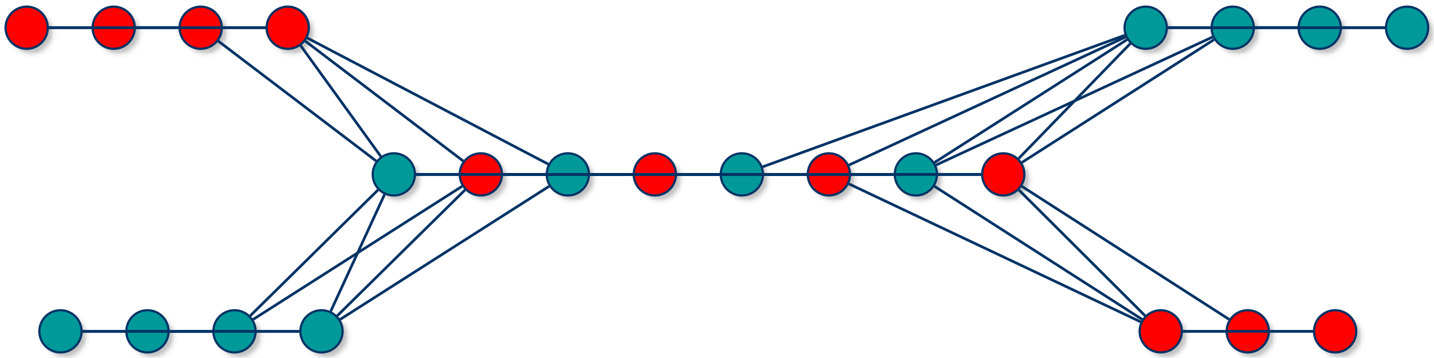
In practice, error correction removes up to 98% of the errors

2. Merge Reads into Contigs

- Overlap graph:
  - Nodes: reads $r_1 \ldots\ldots r_n$
  - Edges: overlaps (r_i, r_j, shift, orientation, score)

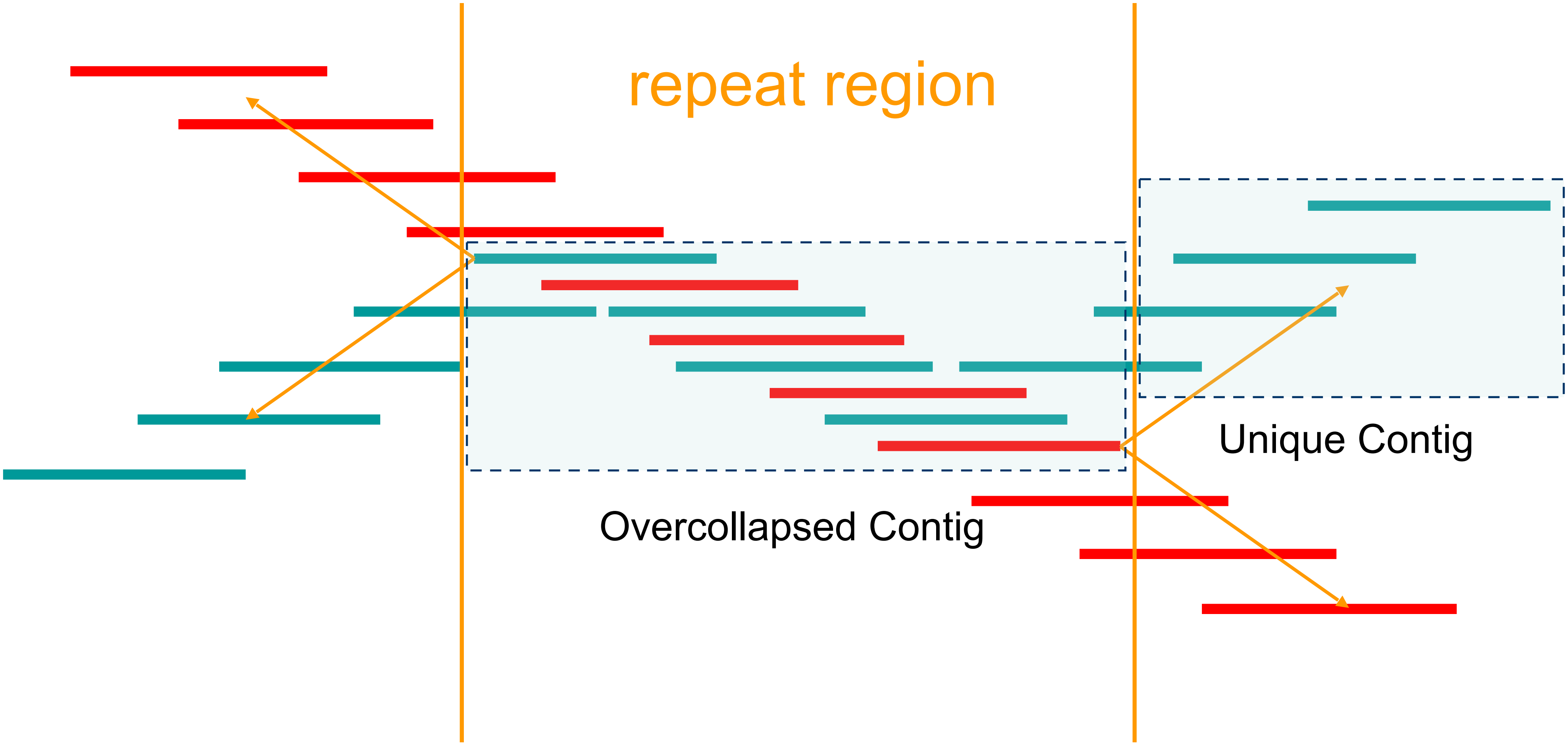


Reads that come from two regions of the genome (blue and red) that contain the same repeat



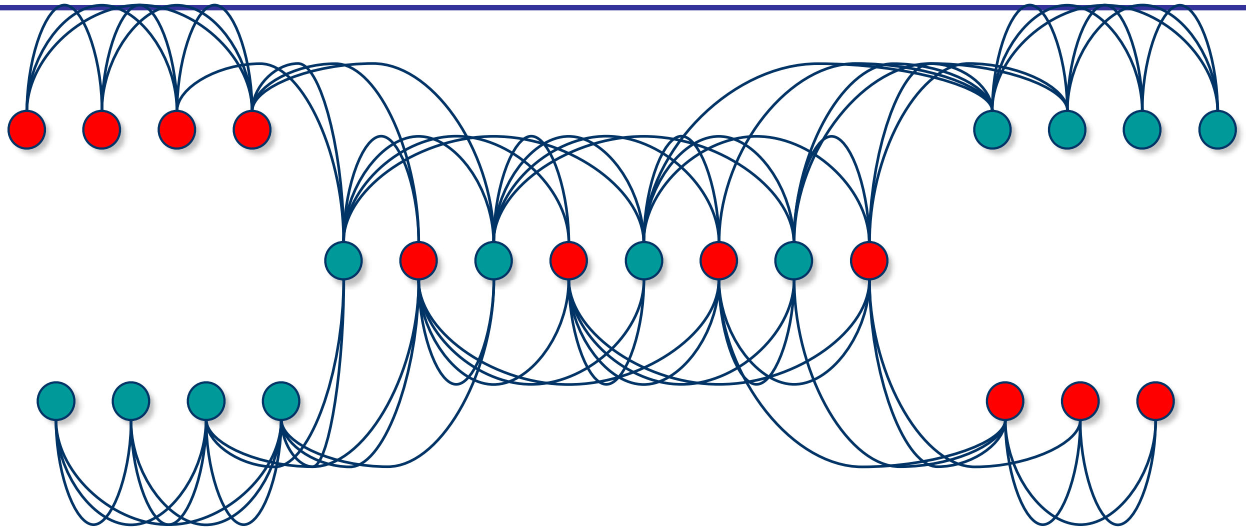
Note:
of course, we don't know the "color" of these nodes

2. Merge Reads into Contigs

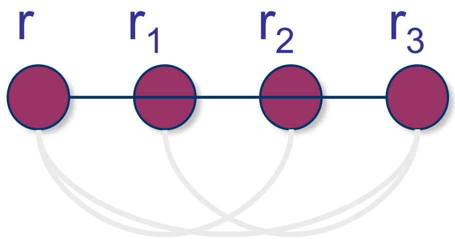


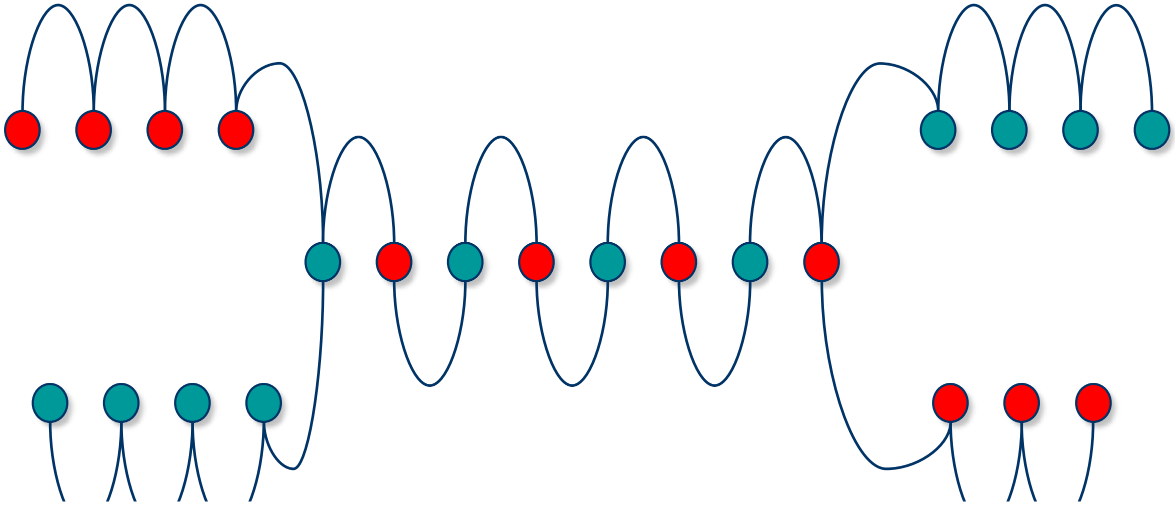
We want to merge reads up to potential repeat boundaries

2. Merge Reads into Contigs

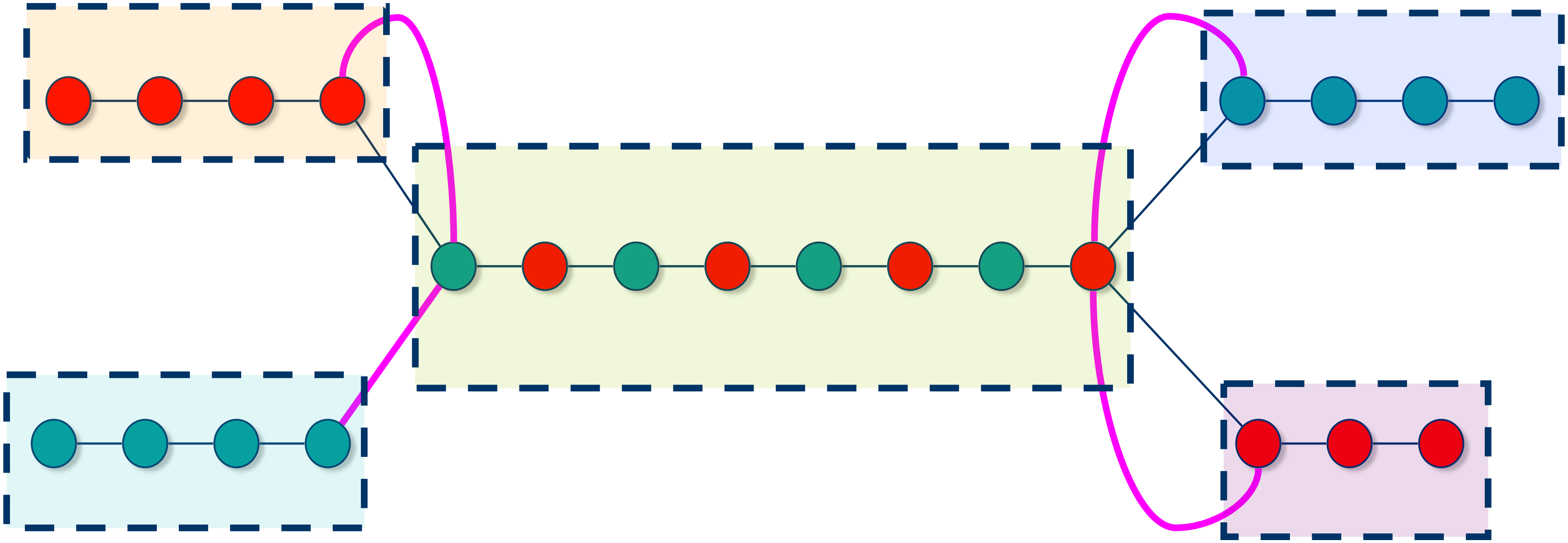


- Remove transitively inferable overlaps
  - If read r overlaps to the right reads r_1, r_2, and r_1 overlaps r_2, then (r, r_2) can be inferred by (r, r_1) and (r_1, r_2)





2. Merge Reads into Contigs



Repeats, errors, and contig lengths

- Repeats shorter than read length are easily resolved
  - Read that spans across a repeat disambiguates order of flanking regions

- Repeats with more base pair diffs than sequencing error rate are OK
  - We throw overlaps between two reads in different copies of the repeat

- To make the genome **appear** less repetitive, try to:

  - Increase read length
  - Decrease sequencing error rate

Role of error correction:

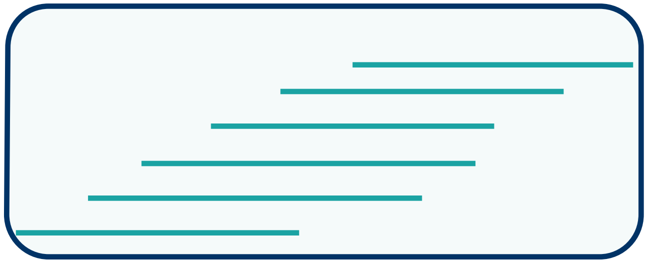
Discards up to 98% of single-letter sequencing errors

decreases error rate

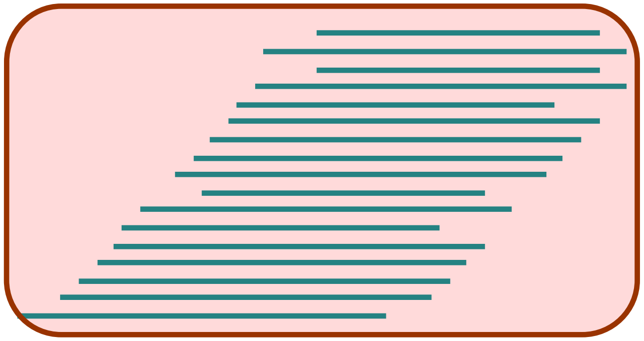
⇒ decreases effective repeat content

⇒ increases contig length

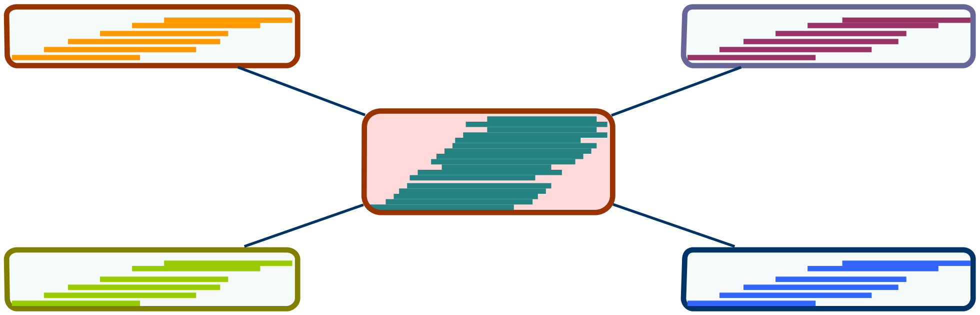
3. Link Contigs into Supercontigs



Normal density



Too dense
⇒ Overcollapsed

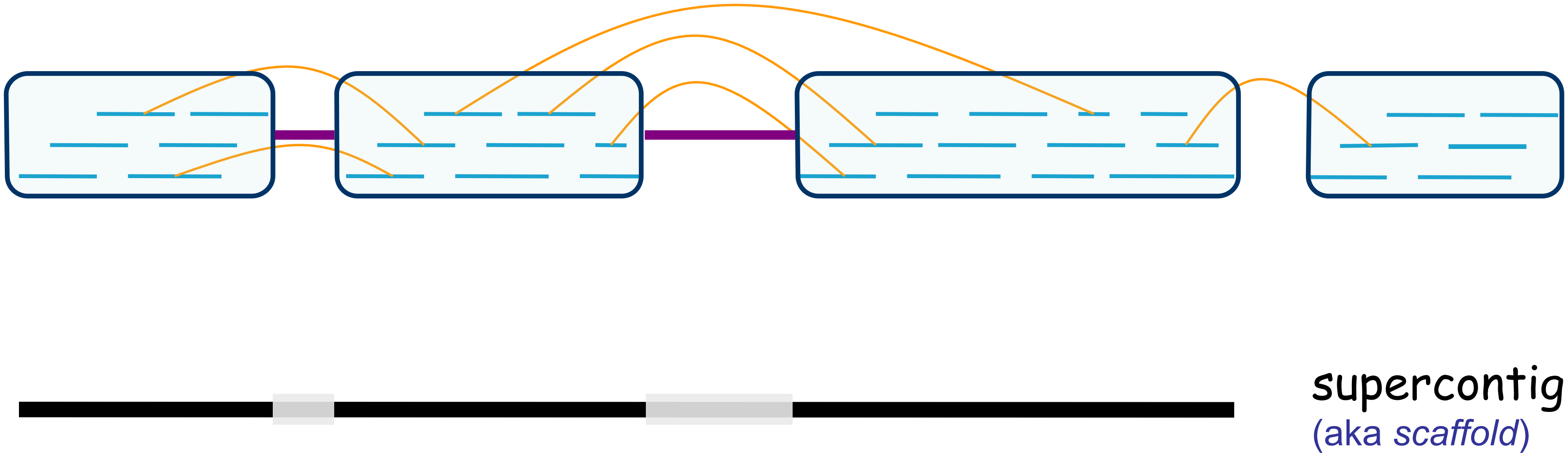


Inconsistent links
⇒ Overcollapsed?

3. Link Contigs into Supercontigs

Find all links between unique contigs

Connect contigs incrementally, if ≥ 2 forward-reverse links

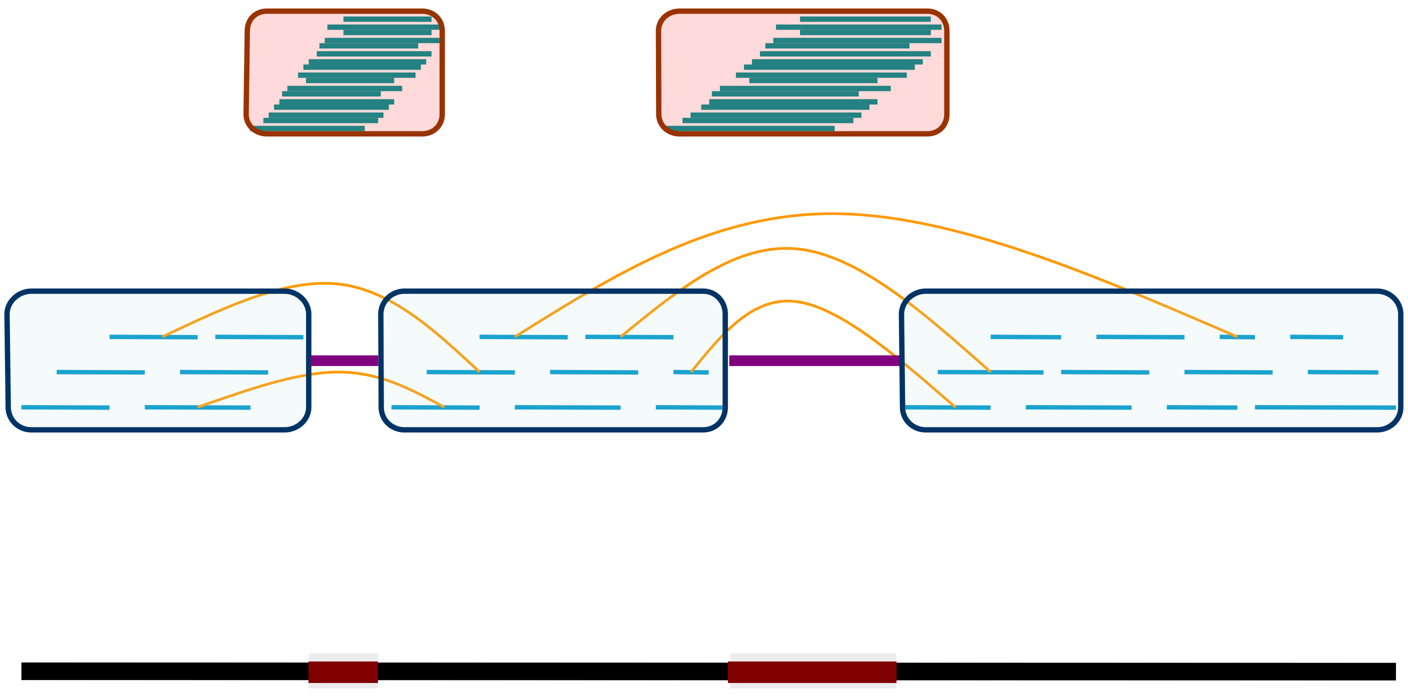


3. Link Contigs into Supercontigs

Fill gaps in supercontigs with paths of repeat contigs

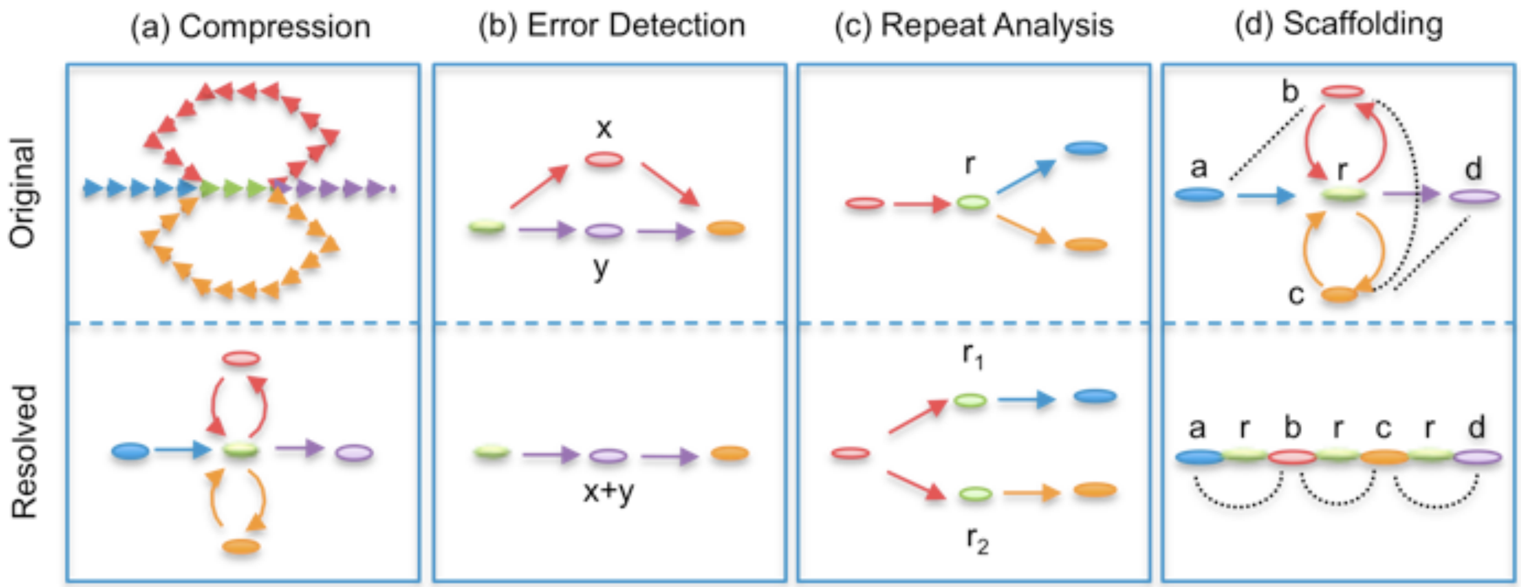
Complex algorithmic step

- Exponential number of paths
- Forward-reverse links

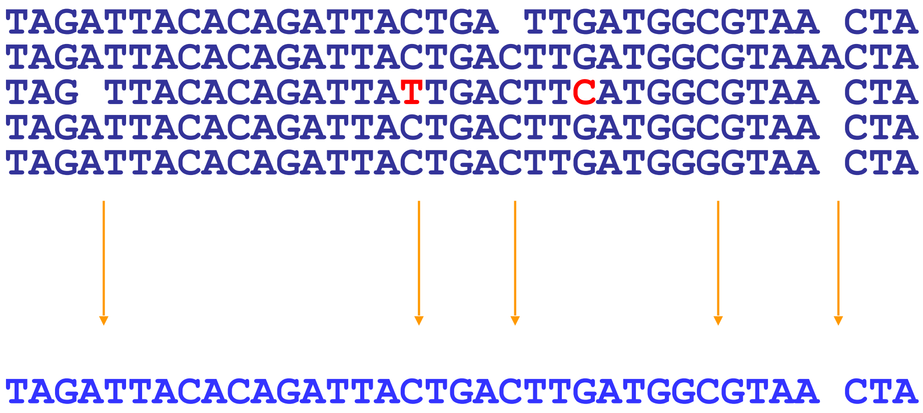


De Brujin Graph formulation

- Given sequence $x_1 \ldots x_N$, k-mer length k,
  Graph of 4^k vertices,
  Edges between words with (k-1)-long overlap



4. Derive Consensus Sequence



```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGGGTAA CTA

TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

Derive multiple alignment from pairwise read alignments

Derive each consensus base by weighted voting

(Alternative: take maximum-quality letter)