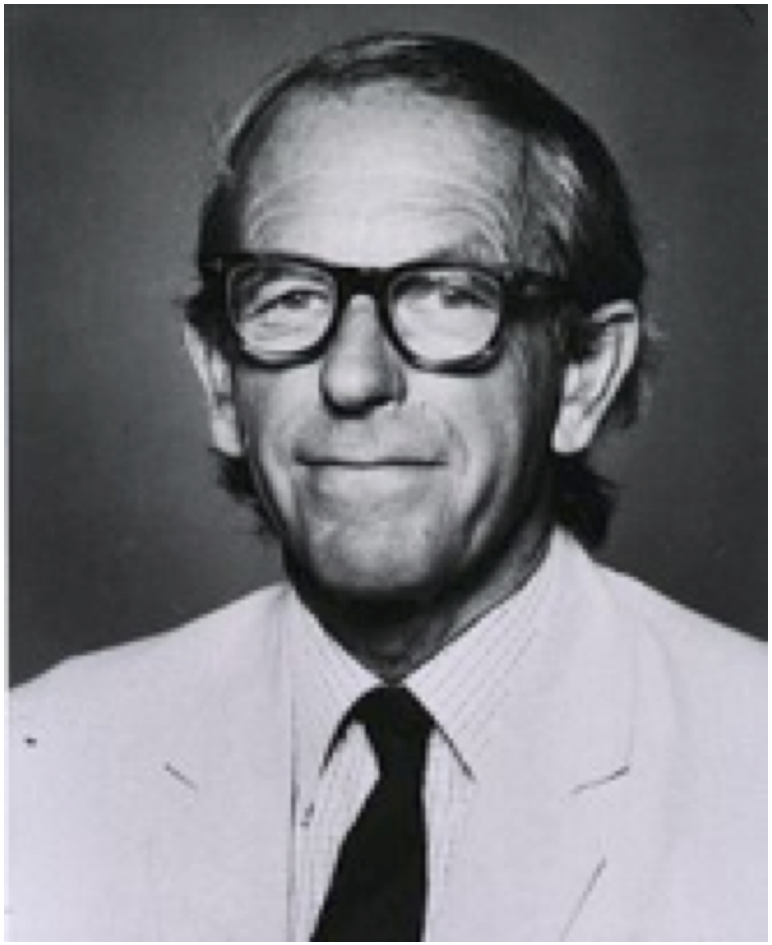
Alignment introduction

Mikhail Dozmorov

Spring 2018

Protein sequencing

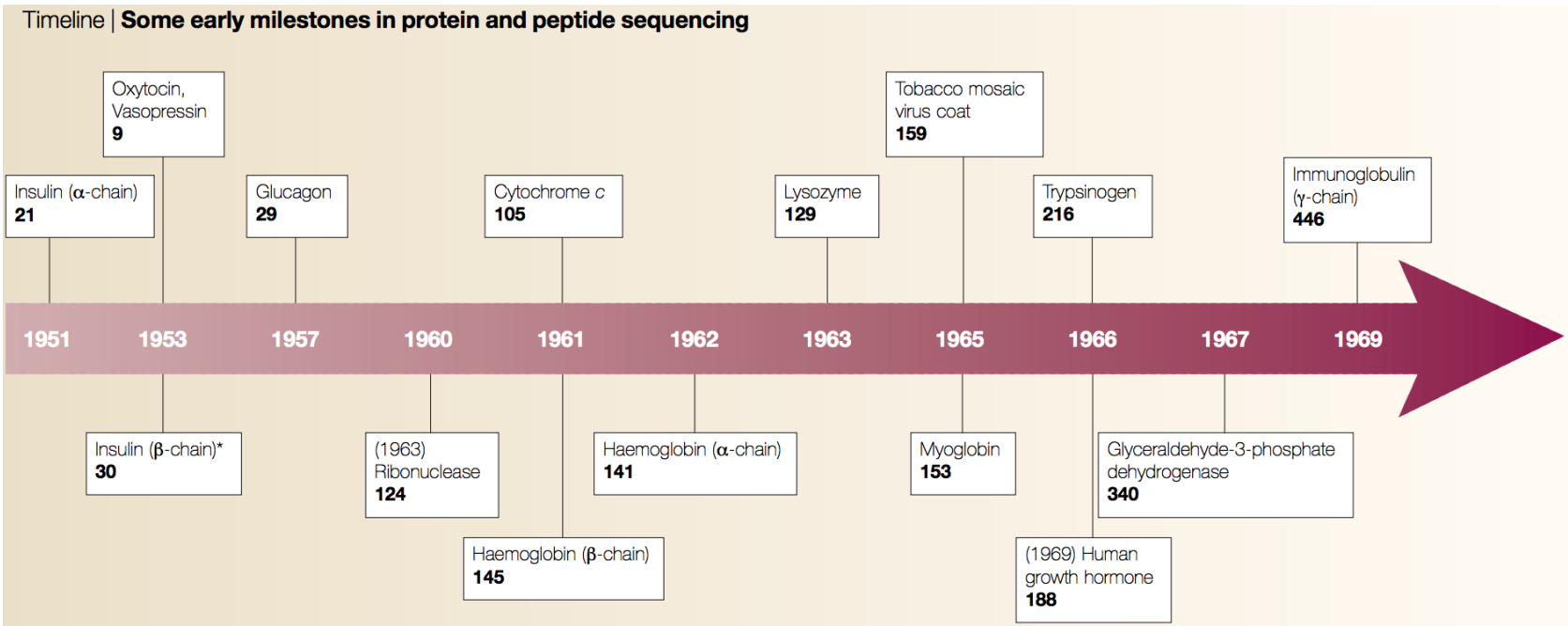
- Fred Sanger and colleagues sequenced Insulin, the first complete protein sequence from 1945-1955
- Established that every protein had a characteristic primary structure
- Moore and Stein developed semi-automated sequencing techniques that transformed protein sequencing



Frederick Sanger. 1958 - his first Nobel Prize
https://onlinelibrarystatic.wiley.com/store/10.1002/pro.5560020715/asset/5560020715_ftp.pdf

1960 - the dawn of computational biology

- Expanding collection of amino acid sequences in the 1960s
- Need for computational power to answer questions and study protein biology
- Scarcity of academic computers was no longer a major problem

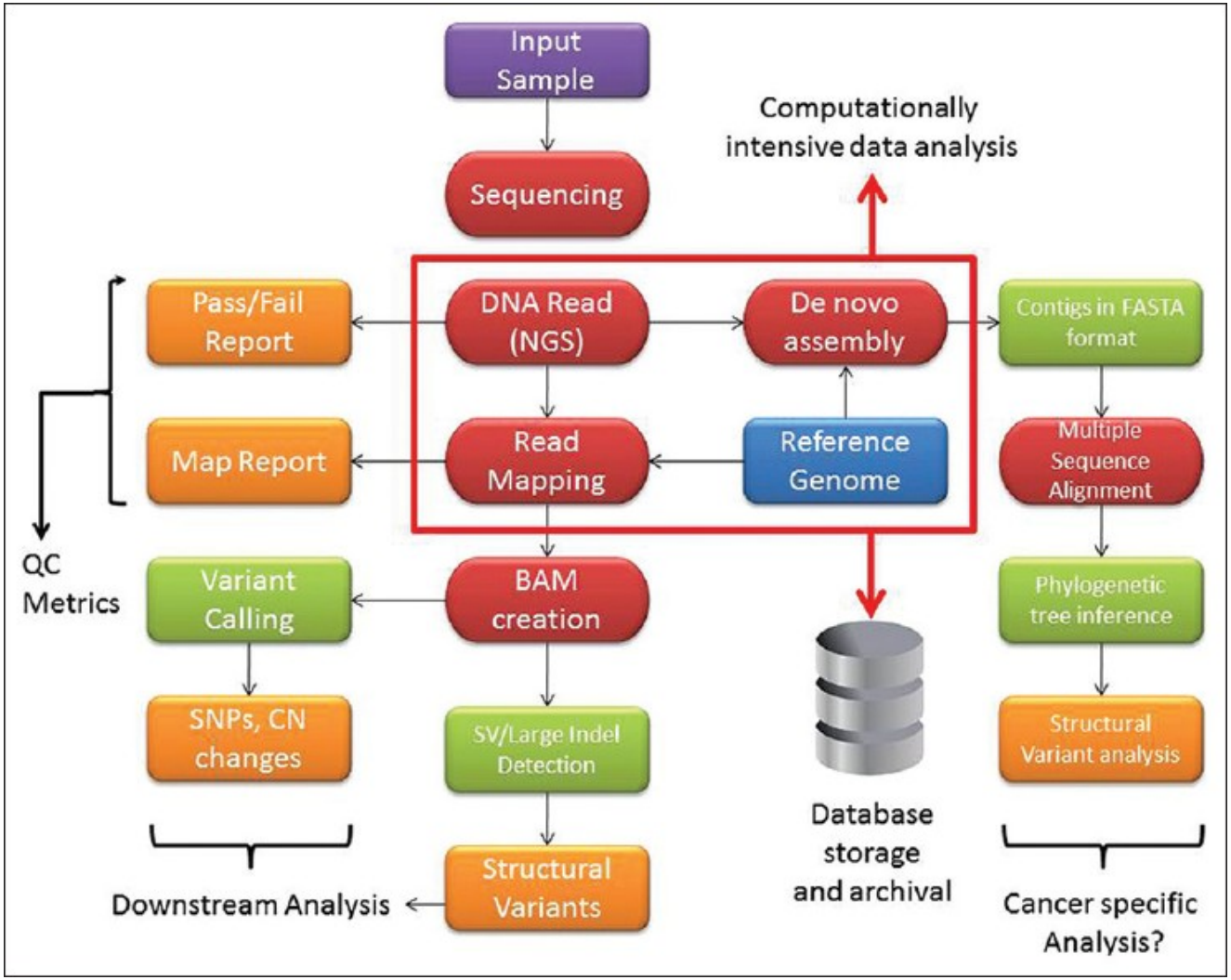


Joel Hagen, "The origins of bioinformatics", NRG, Dec. 2000.
https://www.nature.com/nrg/journal/v1/n3/full/nrg1200_231a.html

Pioneer of Comp. Biology - Margeret Dayhoff

- Trained in math and quantum chemistry
- Associate director of the newly-formed National Biomedical Research Foundation
- Wrote seminal FORTRAN programs to derive amino acids sequences by using partial overlaps of fragmented amino acid sequences.
- PAM (Point accepted mutation) matrices
- Realized the applications to nucleic acids and gene sequences.

The genomics workflow



http://www.jpathinformatics.org/viewimage.asp?img=JPatholInform_2012_3_1_40_103013_u3.jpg

Alignment goals

Alignment - the process by which we discover how or where the read sequence is similar to the reference sequence. Finding best match of the read sequence within the reference sequence.

- The human reference genome is big and complex (~3.2 billion bases)
- Sequencing data is big and complex (~1 billion short reads/run)
- Must find a home to each short read in the reference genome

Alignment goals

Take a read:

```
CTCAAACTCCTGACCTTTGGTGATCCACCCGCCTNGGCCTTC
```

And a reference sequence:

```
>MT dna:chromosome chromosome:GRCh37:MT:1:16569:1
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTT
CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTC
GCAGTATCTGTCTTTGATTCCTGCCTCATCCTATTATTTATCGCACCTACGTTCAATATT
ACAGGCGAACATACTTACTAAAGTGTGTTAATTAATTAATGCTTGTAGGACATAATAATA
ACAATTGAATGTCTGCACAGCCACTTTCCACACAGACATCATAACAAAAAATTTCCACCA
AACCCCCCCTCCCCCGCTTCTGGCCACAGCACTTAAACACATCTCTGCCAAACCCCAAAA
ACAAAGAACCCTAACACCAGCCTAACCAGATTTCAAATTTTATCTTTTGGCGGTATGCAC
TTTTAACAGTCACCCCCCAACTAACACATTATTTTCCCCTCCCACTCCCATACTACTAAT
CTCATCAATACAACCCCCGCCCATCCTACCCAGCACACACACACCGCTGCTAACCCCATA
CCCCGAACCAACCAAACCCCAAACACACCCCCCACACTTTATCTACCTTACCTCCTCAAA
GCAATACACTGACCCGCTCAAACTCCTGGATTTTGGATCCACCCAGCGCCTTGGCCTAAA
CTAGCCTTTCTATTAGCTCTTAGTAAGATTACACATGCAAGCATCCCCGTTCCAGTGAGT
TCACCCTCTAAATCACCACGATCAAAAGGAACAAGCATCAAGCACGCAGCAATGCAGCTC
AAAACGCTTAGCCTAGCCACACCCCCACGGGAAACAGCAGTGATTAACCTTTAGCAATAA
ACGAAAGTTTAACTAAGCTATACTAACCCCAGGGTTGGTCAATTTCGTGCCAGCCACCGC
GGTCACACGATTAACCCAAGTCAATAGAAGCCGGCGTAAAGAGTGTTTTAGATCACCCCC
TCCCCAATAAAGCTAAAACTCACCTGAGTTGTAAAAAACTCCAGTTGACACAAAATAGAC
TACGAAAGTGGCTTTAACATATCTGAACACACAATAGCTAAGACCCAAACTGGGATTAGA
TACCCCACTATGCTTAGCCCTAAACCTCAACAGTTAAATCAACAAAACTGCTCGCCAGAA
CACTACGAGCCACAGCTTAAAACTCAAAGGACCTGGCGGTGCTTCATATCCCTCTAGAGG
AGCCTGTTCTGTAATCGATAAACCCCGATCAACCTCACCACCTCTTGCTCAGCCTATATA
CCGCCATCTTCAGCAAACCCTGATGAAGGCTACAAAGTAAGCGCAAGTACCCACGTAAAG
ACGTTAGGTCAAGGTGTAGCCCATGAGGTGGCAAGAAATGGGCTACATTTTCTACCCCAG
AAAACTACGATAGCCCTTATGAAACTTAAGGGTCGAAGGTGGATTTAGCAGTAAACTAAG
AGTAGAGTGCTTAGTTGAACAGGGCCCTGAAGCGCGTACACACCGCCCGTCACCCTCCTC
AAGTATACTTCAAAGGACATTTAACTAAAAGGGCTACGGCATTTATATAGAGGAGACAAGT
CGTAACCTCAAACTCCTGCCTTTGGTGATCCACCCGCCTTGGCCTACCTGCATAATGAAG
AAGCACCCAACTTACACTTAGGAGATTTCAACTTAACTTGACCGCTCTGAGCTAAACCTA
GCCCCAAACCCACTCCACCTTACTACCAGACAACCTTAGCCAAACCATTTACCCAAATAA
AGTATAGGCGATAGAAATTGAAACCTGGCGCAATAGATATAGTACCGCAAGGGAAAGATG
AAAAATTATAACCAAGCATAATATAGCAAGGACTAACCCCTATACCTTCTGCATAATGAA
```

How do we determine the read's point of origin with respect to the reference?

Hypothesis 1:

Read

```
CTCAAAGACCTGACCTTTGGTGATCCACCC-----GCCTNGGCCTTC
||||||  ||||   ||||  ||||||||||     |||| |||||
CTCAAACTCCTGGATTTTG--GATCCACCCAGCTGGCCTTGGCCTAA
```

Reference

Hypothesis 2:

Read

```
CTCAAACTCCTGACCTTTGGTGATCCACCCGCCTNGGCCTTC
|||||||||||| ||||||||||||||||||||| |||||| |
CTCAAACTCCTG-CCTTTGGTGATCCACCCGCCTTGGCCTAC
```

Reference

Which hypothesis is better?

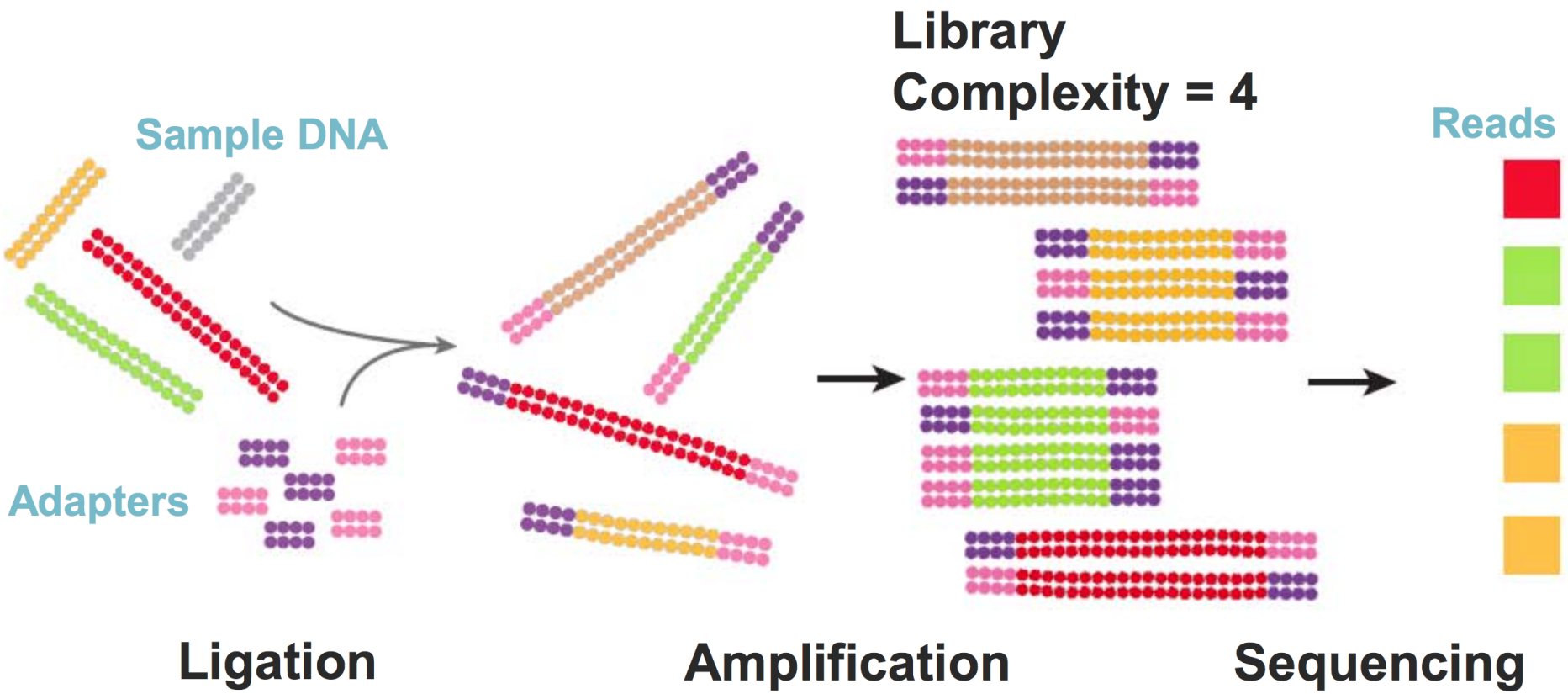
Say hypothesis 2 is correct. Why are there still mismatches and gaps?

Sequencing coverage

- Average number of reads covering genomic bases
- If the genome is 100 Mbp, should we sequence 1M x 100bp reads?

Library complexity

- Library complexity is the number of unique molecules in the "library" that is sampled by finite sequencing



Modeling approach

- Assume we have C unique molecules in the library and we obtain N sequencing reads
- The probability distribution of the number of times we sequence a particular molecule is binomial (individual success probability $p = 1/C$, N trials in total)
- Assume Poisson sampling as a tractable approximation (rate $\lambda = N/C$)
- Finally, truncate the Poisson process: we only see events that happened between L and R times (we don't know how many molecules were observed 0 times)

Poisson Distribution

- The probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event.
- Resembles a normal distribution, but over the positive values, and with only a single parameter.

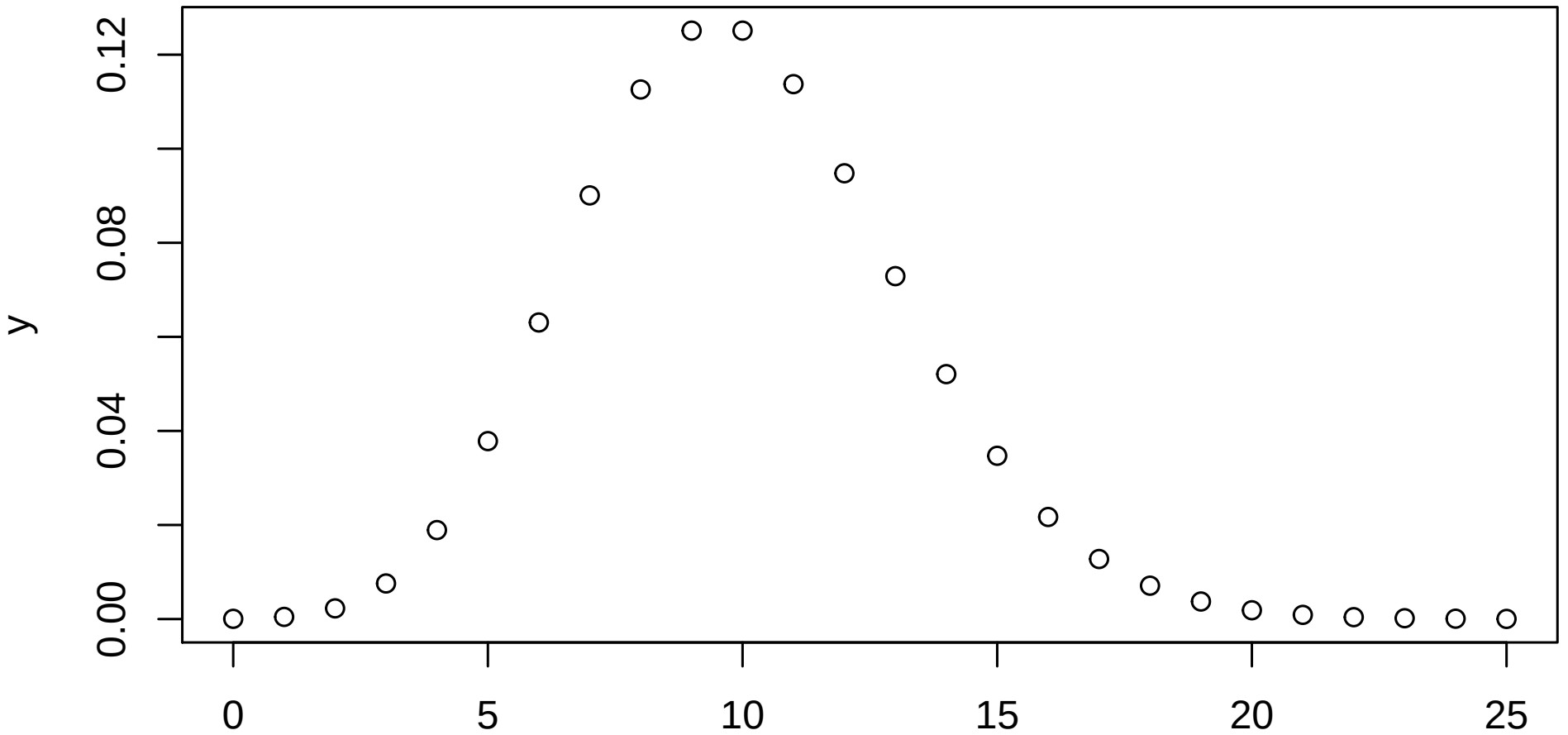
Key properties

- The standard deviation is the square root of the mean.
- For mean > 5, well approximated by a normal distribution

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

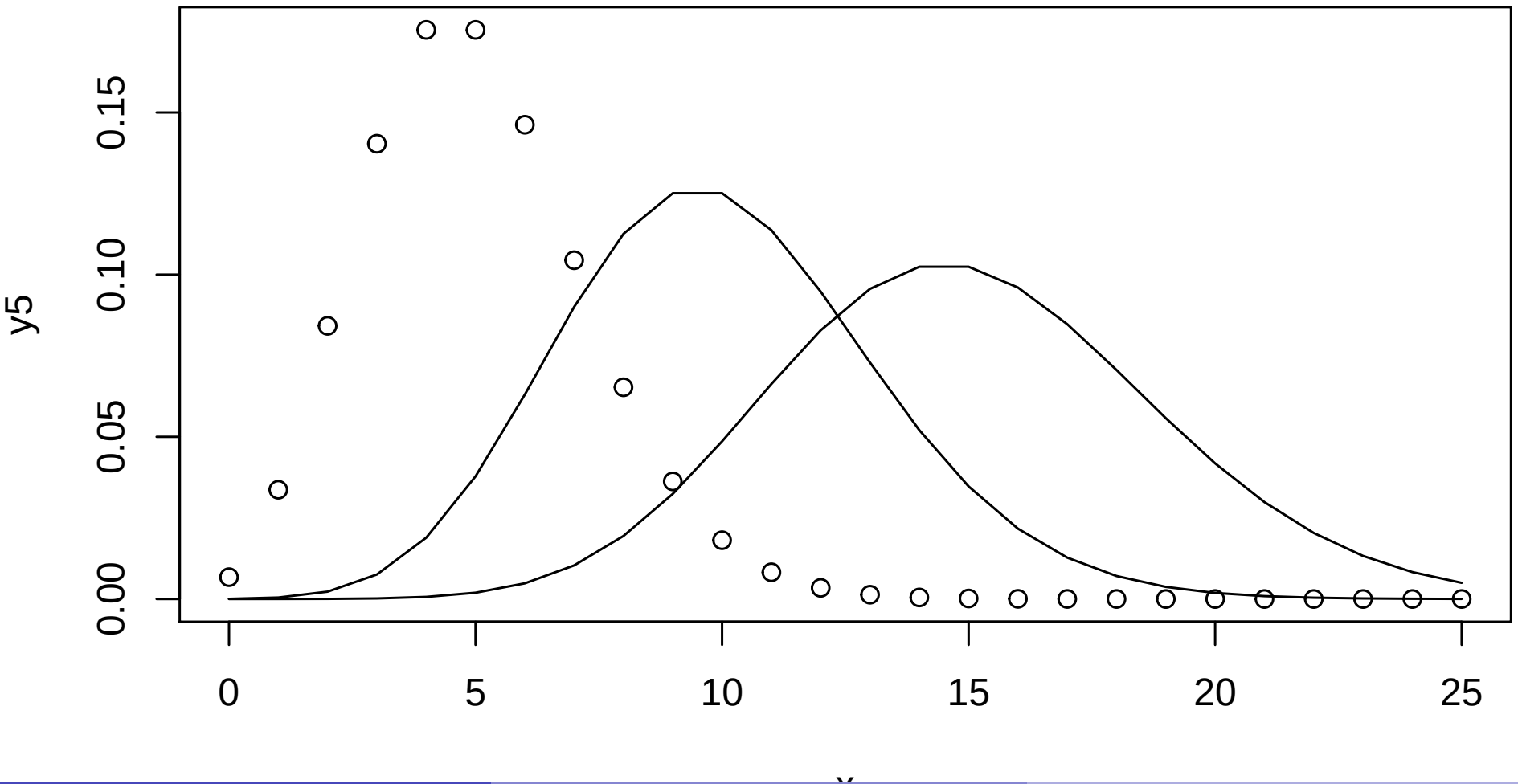
Poisson discribution

```
x <- seq(0, 25, 1)
y <- dpois(x, 10)
plot(x, y)
```



Poisson discribution

```
y5  <- dpois(x, 5); y10 <- dpois(x, 10); y15 <- dpois(x, 15)
plot(x, y5, col = 1); lines(x, y10); lines(x, y15)
```



Estimating library complexity with a Poisson model

- For Poisson sampling, we can write the (truncated) distribution over x_i the times we sequence the i^{th} molecule as:

$$Pr(x_i|\lambda) = \frac{1}{K_{L,R}(\lambda)} * \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

$$K_{L,R}(\lambda) = \sum_{x=L}^{R} Pr(x_i|\lambda)$$

  (The probability is 0 if x_i is less than L or greater than R)
- We can estimate the maximum likelihood rate parameter λ from a vector of observations x

Maximum likelihood library size

$$K_{L,R}(\lambda) = \sum_{x=L}^{R} Pr(x_i|\lambda)$$

- M unique sequences observed, maximum likelihood library size is

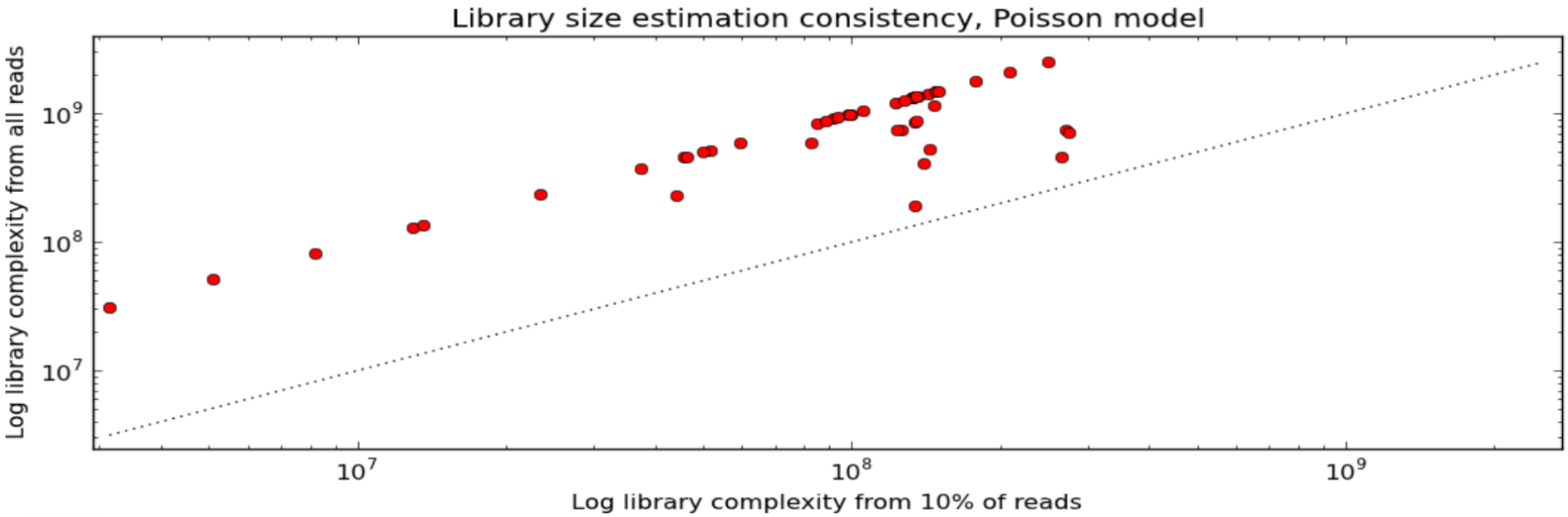
$$\hat{C} = \frac{M}{K_{L,R}(\lambda)}$$

- Approximate solution

$$\hat{C} = \frac{M}{1 - Poisson(0, \lambda)}$$

Problem with Poisson distribution

- Poisson Library Complexity model 150 '1000 Genome' Datasets
- Estimate library complexity from 10% of uniformly sampled reads vs. from all reads

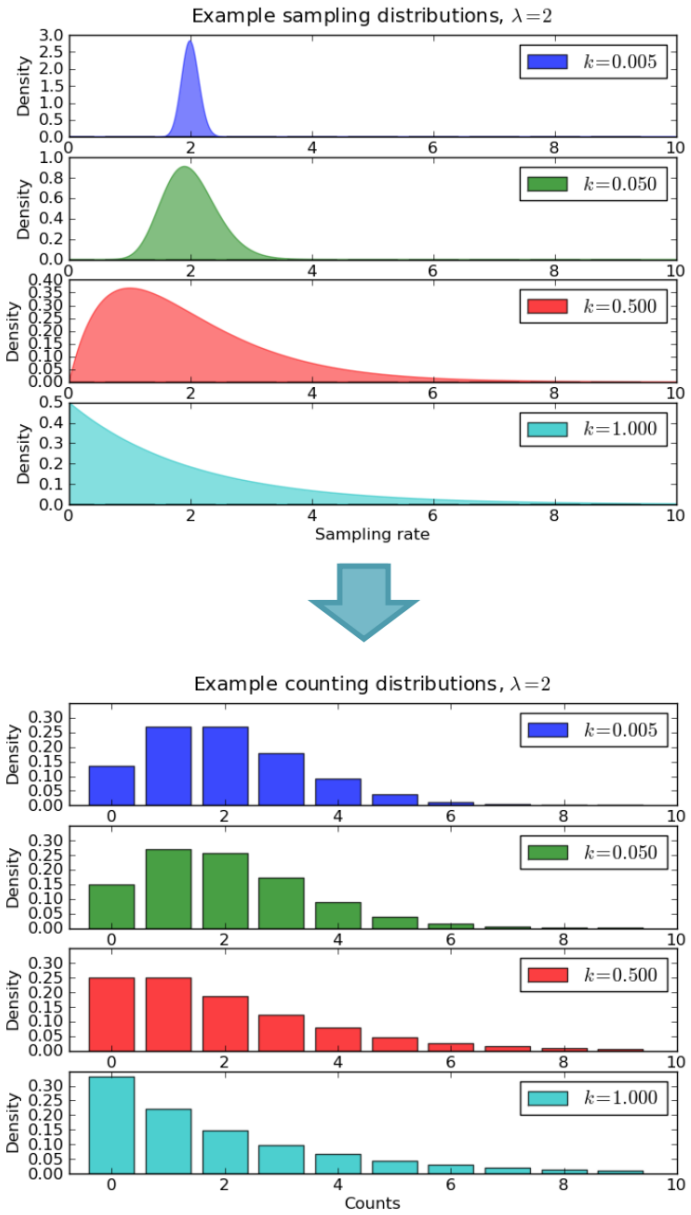


- Poisson $\lambda = Mean = Variance$

- Gamma distributed sampling rates describe the entire population (library preparation)

- Poisson sampling to form a smaller sample (sequencing)

- Negative binomial distribution characterizes the resulting occurrence histogram



The gamma distribution is a "conjugate prior" for the Poission distribution

$$Poisson(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$Gamma(x,\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

$$NB(y;\alpha,\beta) = \int_0^\infty Poisson(y;x)\, Gamma(x;\alpha,\beta)\, dx$$

Negative Binomial model for sequence occurrences

- C - library complexity (latent, fit to observed data)
- N - number of reads
- M - total number of unique sequences
- $\lambda = N/C$
- k - dispersion (latent, fit to observed data)

$$Pr(x_i, \lambda, k) = NB(x_i|\lambda, k) = NB(x_i|n, p)$$

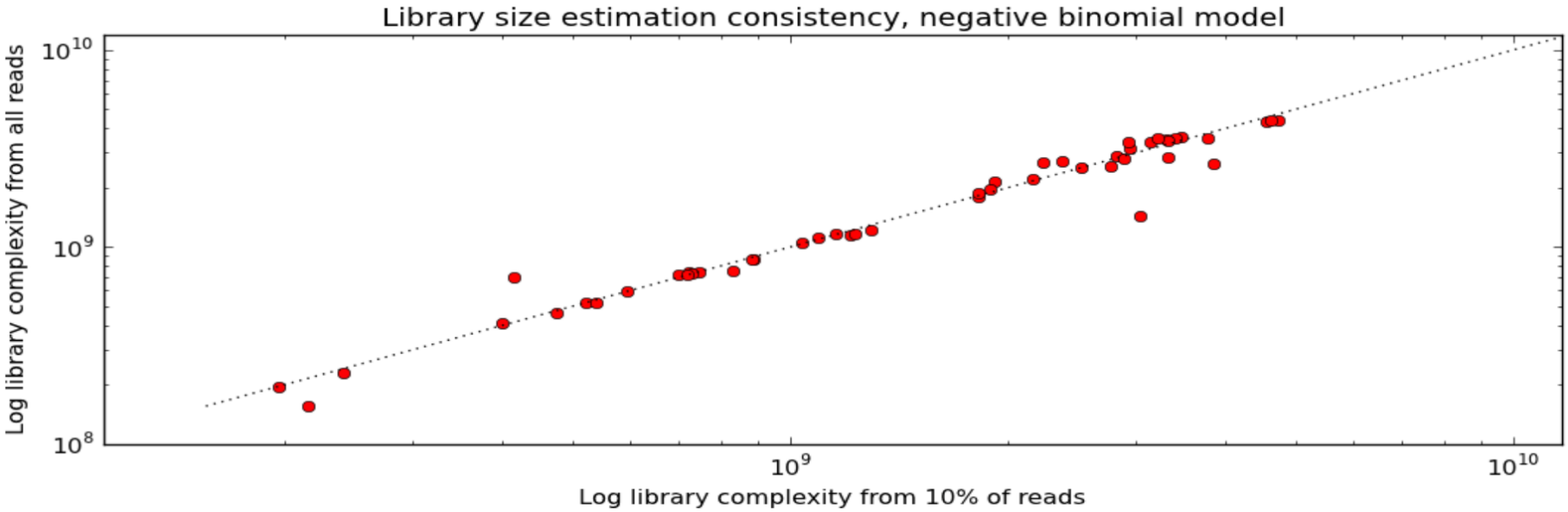
- $p = \lambda/(\lambda + 1/k)$
- $n = 1/k$

Simulation results show that the Gamma Possion works well for non-uniform libraries

- True library complexity: 1M unique molecules
- Vary k (controls sampling rate variance)
- Given 100K reads ($\lambda = 0.1$), assess estimates from both models

- k=0.1 Poisson: 0.93M GP: 0.96M
- k=1 Poisson: 0.52M GP: 1.01M
- k=10 Poisson: 0.12M GP: 1.10M
- k=20 Poisson: 0.07M GP: 0.68M

Negative Binomial Library Complexity better model 150 '1000 Genome' Datasets



- Data are "overdispersed" (variance greater than mean)

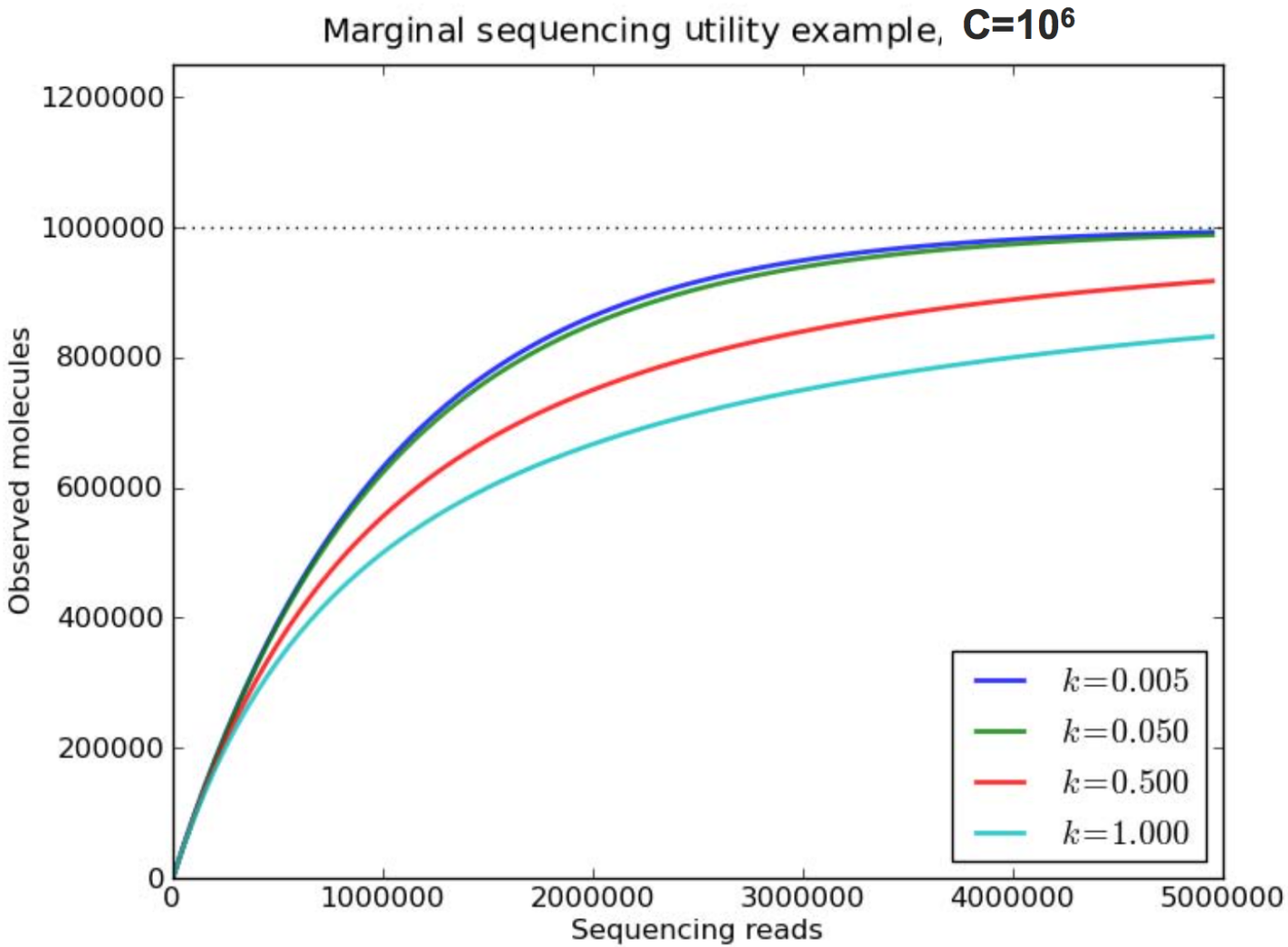
Marginal value of additional sequencing

- C – library complexity (latent – estimated)
- N – number of reads
- M – number of unique sequences

M can be estimated by $(1 - NegativeBinomial(0|\lambda, k)) * C$

- Assume we have r more reads $s = (N + r)/N$
- Replace λ by $s * \lambda$ to estimate M' achieved with r more reads

Marginal value of additional sequencing



Genome Assembly Algorithms

Problem: Exact String Matching

- **Input**: A text string T, where $\|T\| = n$, and a pattern string P, where $\|P\| = m$.
- **Output**: An index i such that $T_{i+j-1} = P_j$ for all $1 \leq j \leq m$, i.e. showing that P is a substring of T.

Analysis

- This algorithm might use only n steps if we are lucky, e.g. $T = aaaaaaaaaaa$, and $P = bbbbbbb$.
- We might need $\sim n \times m$ steps if we are unlucky, e.g. $T = aaaaaaaaaaa$, and $P = aaaaaab$.
- We can't say what happens "in practice", so we settle for a worst case analysis.
- By being more clever, we can reduce the worst case running time to $O(nm)$.
- Certain generalizations won't change this, like stopping after the first occurrence of the pattern.
- Certain other generalizations seem more complicated, like matching with gaps.

Algorithm Complexity

We use the "Big oh" notation to state an upper bound on the number of steps that an algorithm takes in the worst case. Thus the brute force string matching algorithm is $O(nm)$, or takes *quadratic* time.

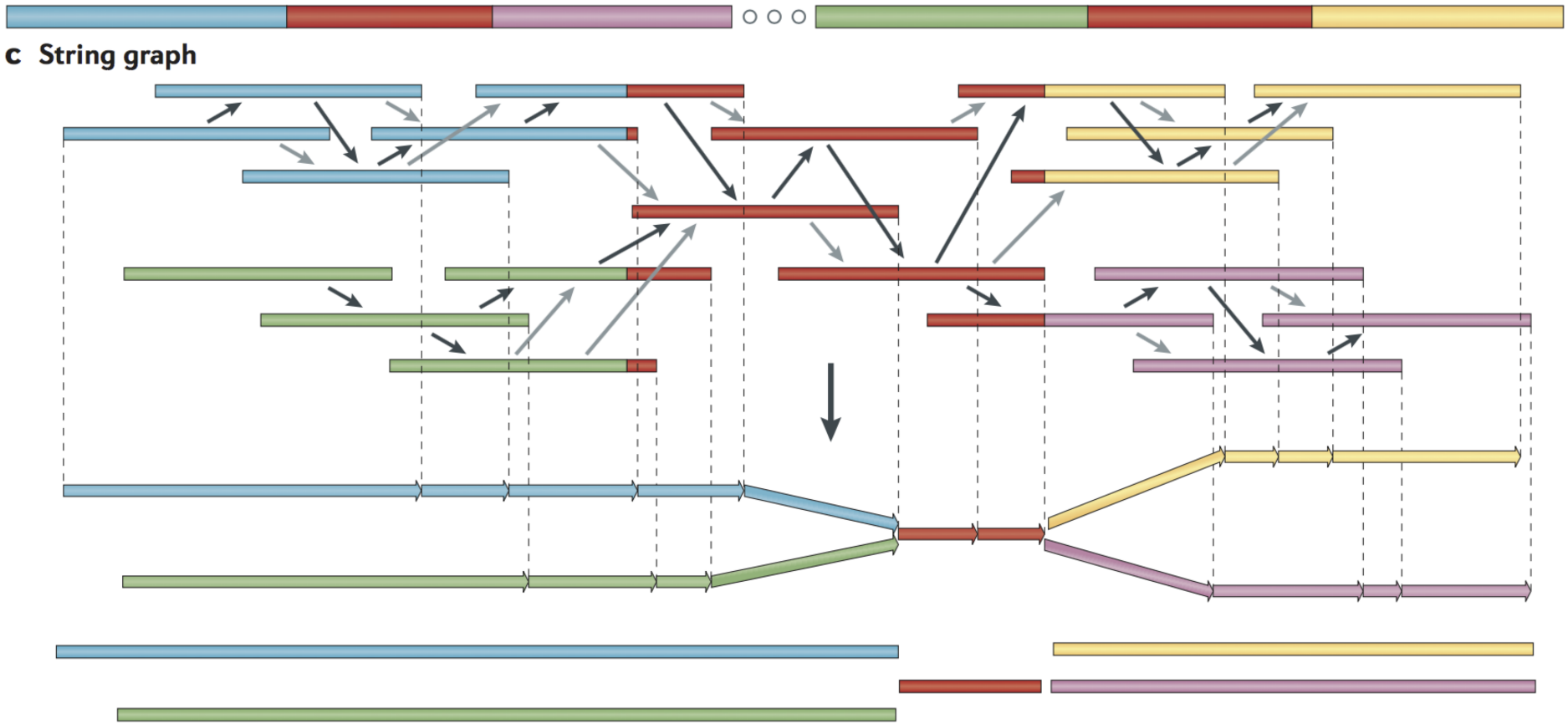
- A *linear* time algorithm, i.e. $O(n+m)$, is fast enough for almost any application.
- A *quadratic* time algorithm is usually fast enough for small problems, but not big ones, since $1000^2 = 1,000,000$ steps is reasonable but $1,000,000^2$ is not.
- An exponential-time algorithm, i.e. $O(2^n)$ or $O(n!)$, can only be fast enough for tiny problems, since 2^{20} and $10!$ are already up to $1,000,000$.
- Unfortunately, for many alignment problems, there is no known polynomial algorithm.
- Even worse, most of these problems can be proven NP-complete, meaning that no such algorithm can exist!

String graph

- Alignments that may be transitively inferred from all pairwise alignments are removed
- A graph is created with a vertex for the endpoint of every read
- Edges are created both for each unaligned interval of a read and for each remaining pairwise overlap
- Vertices connect edges that correspond to the reads that overlap
- When there is allelic variation, alternative paths in the graph are formed

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745987/

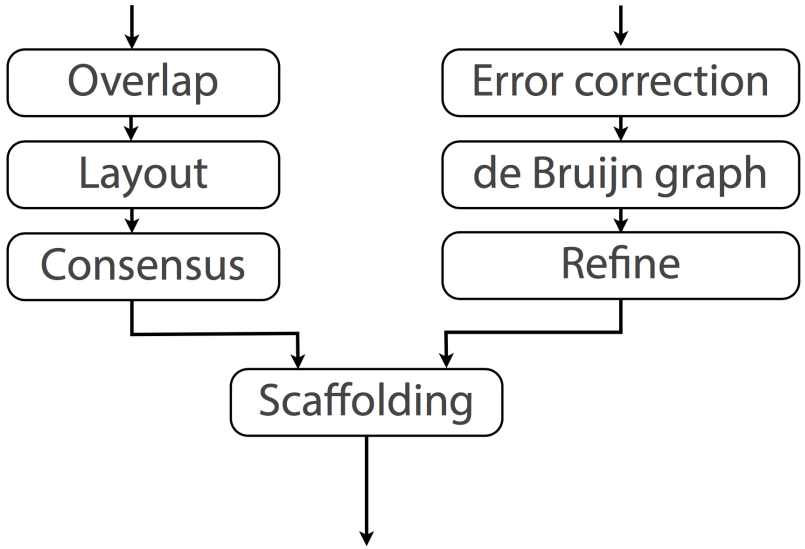
String graph



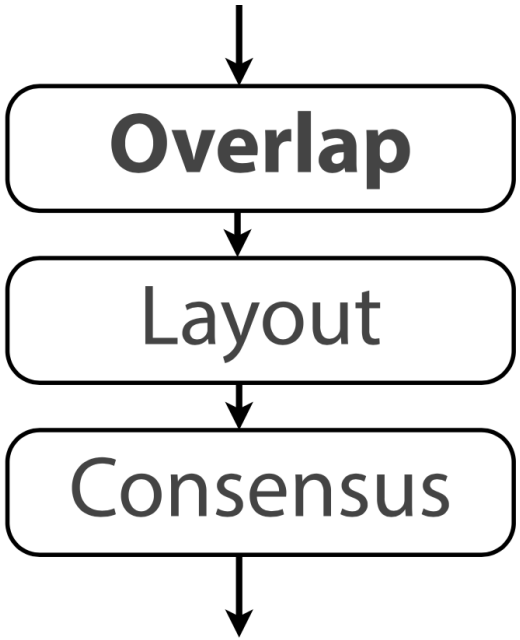
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745987/

Real-world assembly methods

- OLC - Overlap-Layout-Consensus assembly
- DBG - De Bruijn graph assembly
- Both handle unresolvable repeats by essentially leaving them out
- Unresolvable repeats break the assembly into fragments Fragments are contigs (short for contiguous)



Overlap-layout-consensus (OLC)



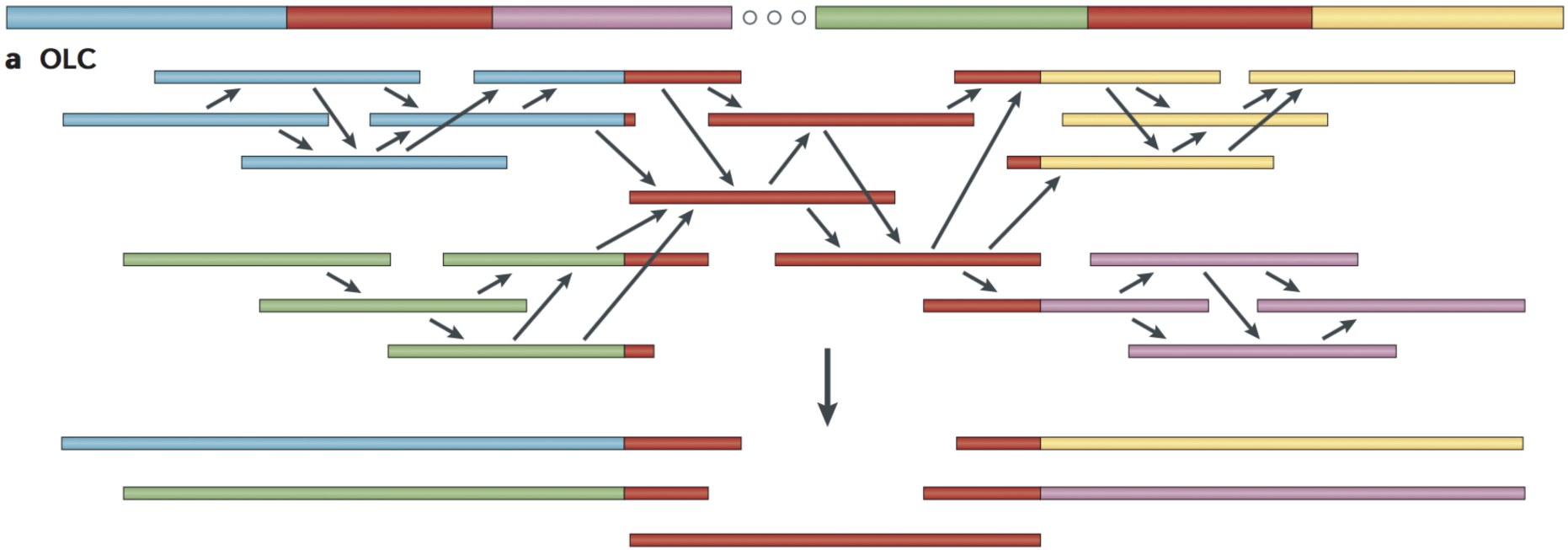
Build overlap graph

Bundle stretches of the overlap graph into *contigs*

Pick most likely nucleotide sequence for each contig

Overlap-layout-consensus (OLC)

- All pairwise alignments (arrows) between reads (solid bars) are detected.
- Reads are merged into contigs (below the vertical arrow) until a read at a repeat boundary (split colour bar) is detected, leading to a repeat that is unresolved and collapsed into a single copy.



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745987/

Overlap graph formulation

- Treat each sequence as a "node"
- Draw an edge between two nodes if there is significant overlap between the two sequences
- Hopefully the contig covers all or large number of sequences, once for each sequence
- In other words, we are looking for Hamiltonian path in the overlap graph
- Pros: straightforward formulation
- Cons: no efficient accurate algorithm; repeats

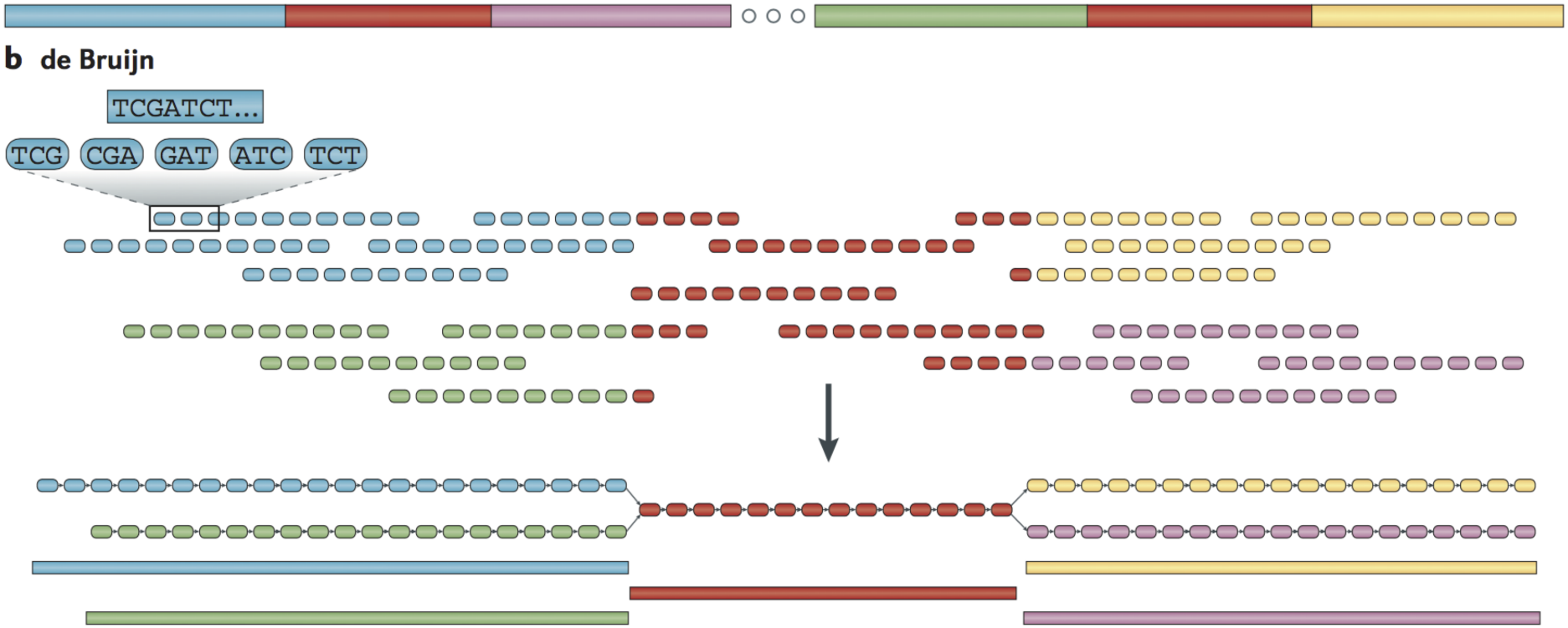
de Bruijn assembly

- Reads are decomposed into overlapping k-mers.
- Identical k-mers are merged and connected by an edge when appearing adjacently in reads.
- Contigs are formed by merging chains of k-mers until repeat boundaries are reached.
- If a k-mer appears in multiple positions (red segment) in the genome, it will fragment assemblies and additional graph operations must be applied to resolve such small repeats.
- The k-mer approach is ideal for short-read data generated by massively parallel sequencing (MPS).

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745987/

de Bruijn assembly

- An example of the decomposition for k = 3 nucleotides is shown, although in practice k ranges between 31 and 200 nucleotides.

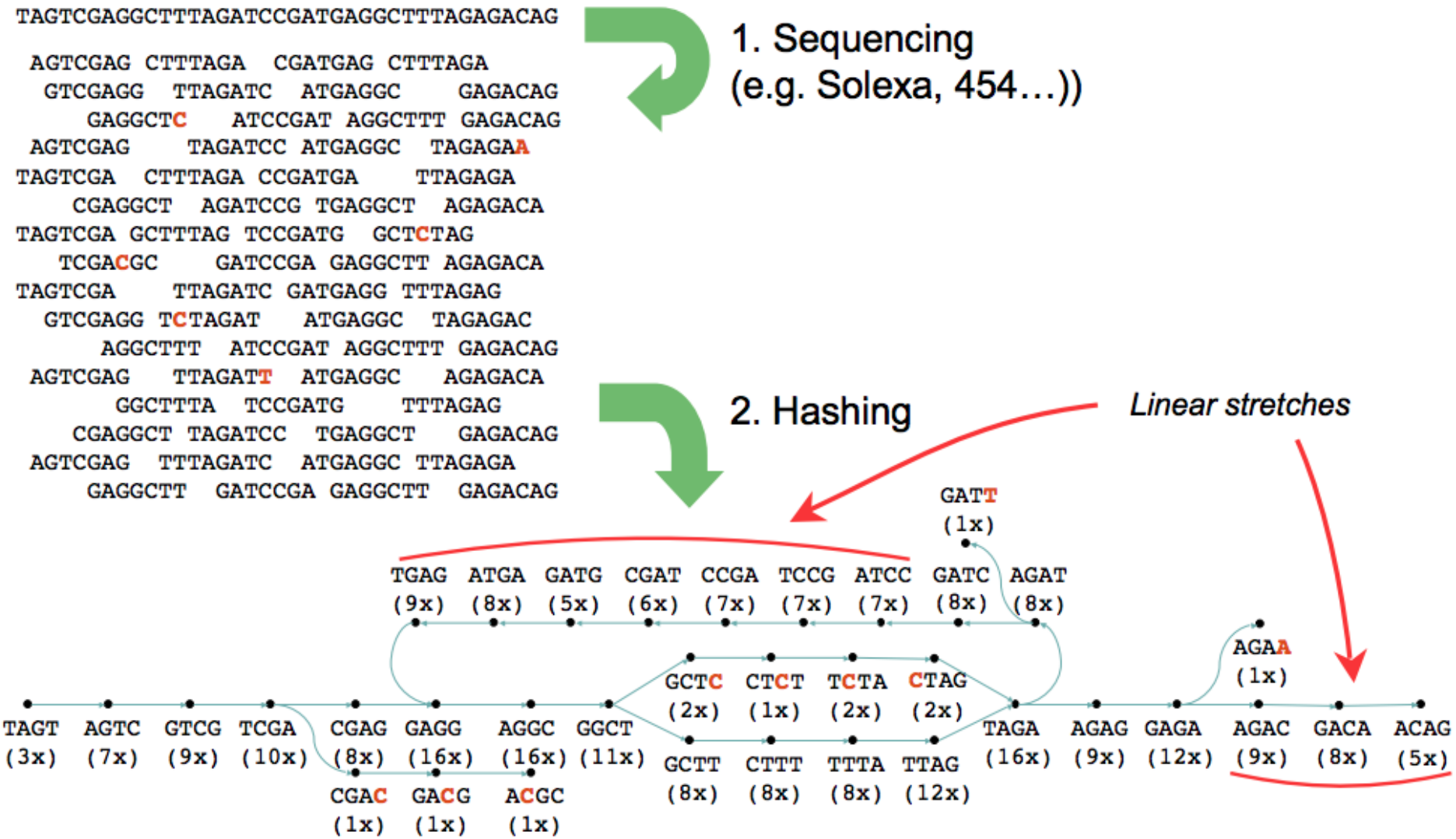


https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745987/

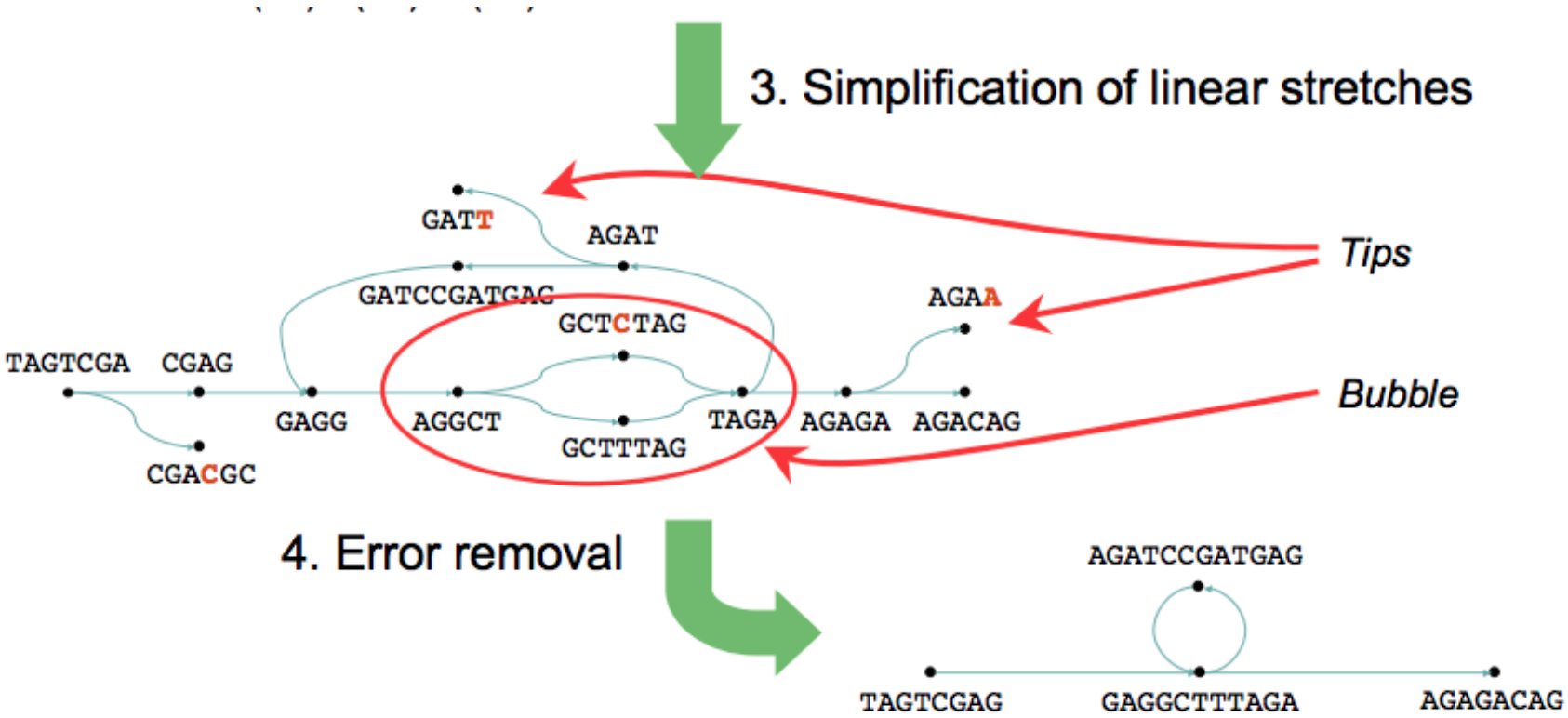
de Bruijn assembly problems

- Erroneous data create three types of graph structures:
  - "tips" due to errors at the edges of reads,
  - "bulges" due to internal read errors or to nearby tips connecting
  - erroneous connections due to cloning errors or to distant merging tips.

Velvet: de novo assembly using very short reads



Velvet: de novo assembly using de Bruijn graph



https://www.ebi.ac.uk/~zerbino/velvet/

Zerbino, Daniel R., and Ewan Birney. "Velvet: Algorithms for de Novo Short Read Assembly Using de Bruijn Graphs." Genome Research 18, no. 5 (May 2008): 821–29. https://doi.org/10.1101/gr.074492.107.

Issues with reference genome sequence

Alignment problems

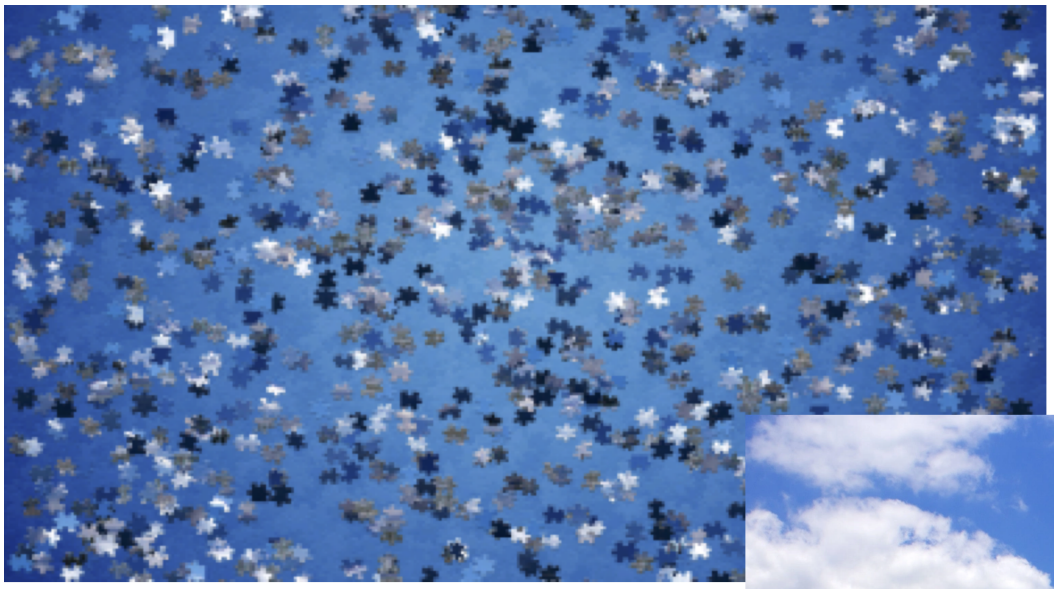
- The genome being sequenced contains genomic variants
- Reads contain two kinds of errors: base substitutions and indels. Base substitutions occur with a frequency from 0.5 – 2%. Indels occur roughly 10 times less frequently
- Strand orientation is unknown
- Computers excel at finding exact matches. Errors should be explicitly handled
- "Fuzzy" pattern matching is much more computationally expensive

Alignment problems

- $>50\%$ of human genome is repeats - a major problem for fragment assembly
- Over 1 million Alu repeats (about 300 bp)
- About 200,000 LINE repeats (1000 bp and longer)

```
taaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaac
cctaacccaaccctaaccctaaccctaaccctaaccctaaccctaacccc
taaccctaaccctaaccctaaccctaacctaaccctaaccctaaccctaa
cccccctaaccctaaccctaaccctaaccccctaaccctaaccctaaaccc
ccctaaaccctaaccctaaccctaaccctaaccctaaccccaaccccaac
cccaaccccaaccccaaccccaaccctaaccccctaaccctaaccctaacc
ctaccctaaccctaaccctaaccctaaccctaaccctaaccccctaacccc
taaccctaaccctaaccctaaccctaaccctaaccctaaccccctaaccct
aaccctaaccctaaccctcgcggtaccctcagccggcccgcccgcccggg
tctgacctgaggagaactgtgctccgccttcagagtaccaccgaaatctg
tgcagaggacaacgcagctccgccctcgcggtgctctccgggtctgtgct
gaggagaacgcaactccgccggcgcaggcgcagagaggcgcgccgcgccg
gcgcaggcgcagacacatgctagcgcgtcggggtggaggcgtggcgcagg
cgcagagaggcgcgccgcgccggcgcaggcgcagagacacatgctaccgc
gtccaggggtggaggcgtggcgcaggcgcagagaggcgcaccgcgccggc
gcaggcgcagagacacatgctagcgcgtccaggggtggaggcgtggcgca
ggcgcagagacgcaagcctacgggcgggggttgggggggcgtgtgttgca
ggagcaaagtcgcacggcgccgggctggggcgggggggagggtggcgccgt
gcacgcgcagaaactcacgtcacggtggcgcggcgcagagacgggtagaa
```

Alignment with repeats



From...

...To

Gaps

- Since we rely on fragment overlaps to identify their position, we must sample sufficient fragments to ensure enough overlaps.
- Let T be the length of the target molecule being sequenced using n random fragments of length l, where we recognize all overlaps of length t or greater.
- The **Lander-Waterman equation** gives the expected number of gaps g as:

$$g = ne^{\frac{-n(l-t)}{T}}$$

Calculations

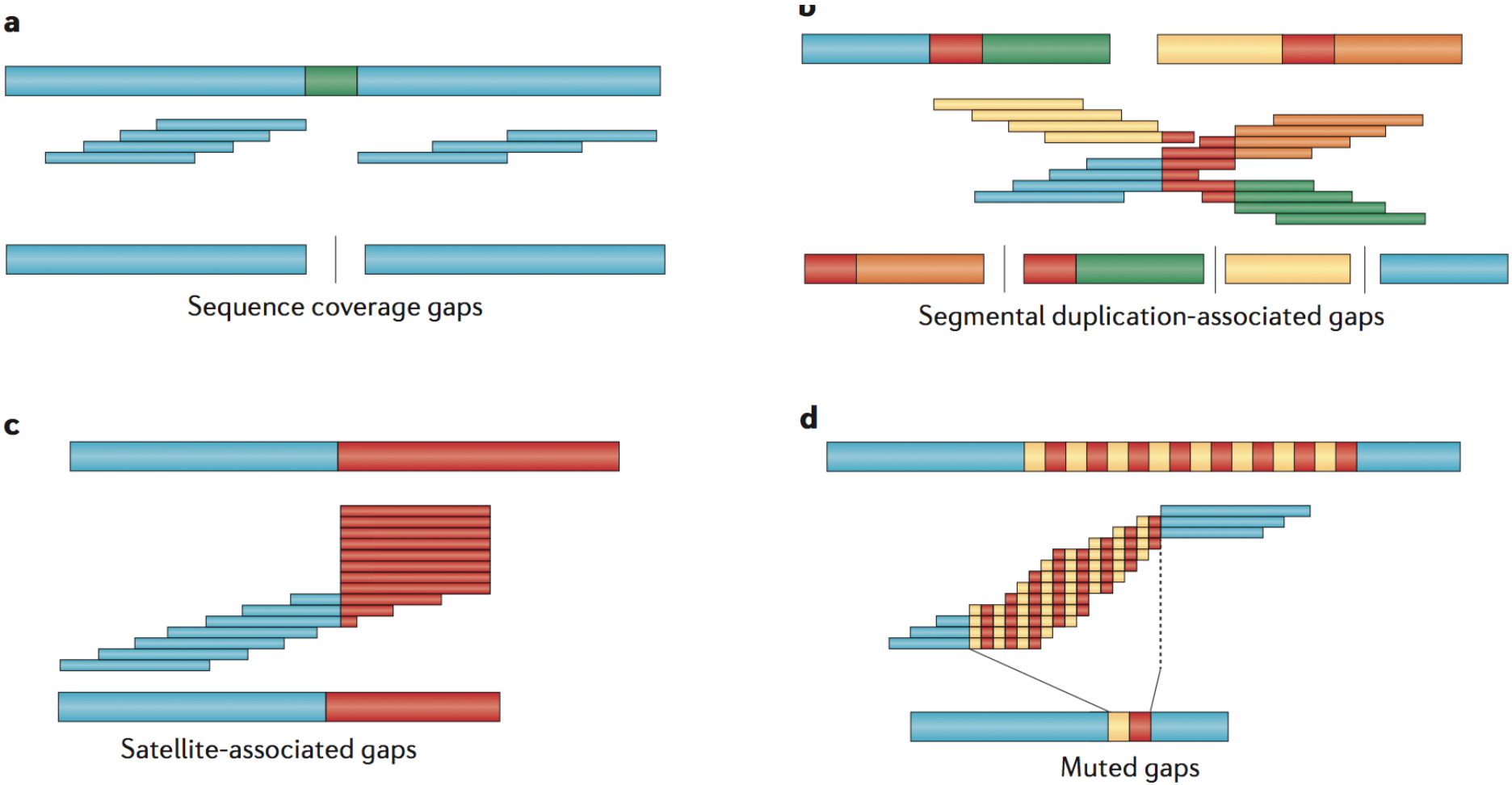
- Suppose we have fragments of length 1. We sequence as many fragments as there is bases. Thus, $T = n$ and each fragment is length 1. The probability p that base i is not sampled is:

$$p = \left(\frac{n-1}{n}\right)^n -> \frac{1}{e}$$

Gaps

- **Sequence-coverage gaps** - Sequencing gaps occur, under the simplest condition, where no sequence reads have been sampled for a particular portion of the genome
- **Segmental duplication-associated gaps** - Over one-third (206/540) of the euchromatic gaps in the human reference genome (GRCh38) are flanked by large, highly identical segmental duplications
- **Satellite-associated gaps** - These include short and long runs of tandem repeats designated as short tandem repeats (STRs; also known as microsatellites), variable number of tandem repeats (VNTRs; also known as macrosatellites) and Mb-sized centromeric satellite repeats
- **Muted gaps** - Muted gaps are defined as regions that are inadvertently closed in an assembly but that actually show additional or different sequences in the vast majority of individuals
- **Allelic variation gaps** - Some regions of a genome also show extraordinary patterns of allelic variation, often reflecting deep evolutionary coalescence

Gaps



http://www.nature.com/nrg/journal/v16/n11/full/nrg3933.html

Coverage

- The coverage of a sequencing project is the ratio of the total sequenced fragment length to the genome length, i.e. nl/T.
- Gaps are very difficult and expensive to close in any sequencing strategy, meaning that very high coverage is necessary to use shotgun sequencing on a large genome.

Evaluating Assemblies

- Coverage is a measure of how deeply a region has been sequenced
- The Lander-Waterman model predicts 8-10 fold coverage is needed to minimze the number of contigs for a 1 Mbp genome
- The **N50** length is a statistics in genomics defined as the shortest contig at which half of the total length of the assembly is made of contigs of that length or greater.
- It is commonly used as a metric to summarize the contiguity of an assembly.

Longer sequencing to complete human genomes

- Human genome is incomplete - ~160 gaps in euchromatin
- ~55% of them have been closed using Oxford Nanopore technology

LETTER

doi:10.1038/nature13907

Resolving the complexity of the human genome using single-molecule sequencing

Mark J. P. Chaisson[1], John Huddleston[1,2], Megan Y. Dennis[1], Peter H. Sudmant[1], Maika Malig[1], Fereydoun Hormozdiari[1], Francesca Antonacci[3], Urvashi Surti[4], Richard Sandstrom[1], Matthew Boitano[5], Jane M. Landolin[5], John A. Stamatoyannopoulos[1], Michael W. Hunkapiller[5], Jonas Korlach[5] & Evan E. Eichler[1,2]

The human genome is arguably the most complete mammalian reference assembly[1–3], yet more than 160 euchromatic gaps remain[4–6] and aspects of its structural variation remain poorly understood ten years after its completion[7–9]. To identify missing sequence and genetic variation, here we sequence and analyse a haploid human genome (CHM1) using single-molecule, real-time DNA sequencing[10]. We close or extend 55% of the remaining interstitial gaps in the human GRCh37 reference genome—78% of which carried long runs of degenerate short tandem repeats, often several kilobases in length, embedded within (G+C)-rich genomic regions. We resolve the complete sequence of 26,079 euchromatic structural variants at the base-pair level, including inversions, complex insertions and long tracts of tandem repeats.

for recruiting additional sequence reads for assembly (Supplementary Information). Using this approach, we closed 50 gaps and extended into 40 others (60 boundaries), adding 398 kb and 721 kb of novel sequence to the genome, respectively (Supplementary Table 4). The closed gaps in the human genome were enriched for simple repeats, long tandem repeats, and high (G+C) content (Fig. 1) but also included novel exons (Supplementary Table 20) and putative regulatory sequences based on DNase I hypersensitivity and chromatin immunoprecipitation followed by high-throughput DNA sequencing (ChIP-seq) analysis (Supplementary Information). We identified a significant 15-fold enrichment of short tandem repeats (STRs) when compared to a random sample ($P < 0.00001$) (Fig. 1a). A total of 78% (39 out of 50) of the closed gap sequences were

https://www.nature.com/nature/journal/v517/n7536/full/nature13907.html

Improving the Human Reference Genome(s)

Reference Genomes Improvement

MGI's commitment to enhancing and diversifying human reference genomes.

Reference Genomes Improvement Details

The Human Genome Project (HGP) produced the human reference genome assembly, a database of DNA sequence that represents an example of a full human genome. When researchers sequence human genomes, they compare, or "align", their results to this reference. While this assembly is one of the most frequently utilized resources in biomedical research, de novo genome assembly remains a significant challenge despite increase in throughput and decrease of sequence cost over the past decade. Alignment of human sequence reads to the reference assembly is a critical aspect of successful data analysis, and several published reports identify regions of the reference assembly that were previously impossible to analyze due to the limitations of the available sequencing technologies, complex genome architecture,

Specific Aims

We plan to identify and resolve issues (misassemblies, sequence errors, and gaps) within the current reference GRCh38. We will add substantial allelic diversity to the reference to facilitate effective analysis of biomedically important regions across the genome. We will accomplish this by completely finishing ("platinum") two genomes (CHM1 and CHM13) and performing targeted finishing ("gold") in additional genomes. We define platinum genome as a contiguous, haplotype-resolved representation of the entire genome. Gold genome is defined as a high-quality, highly contiguous representation of the genome with haplotype resolution of critical regions.

http://genome.wustl.edu/projects/detail/reference-genomes-improvement/

Jain, Miten, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, et al. "Nanopore Sequencing and Assembly of a Human Genome with Ultra-Long Reads." Nature Biotechnology, January 29, 2018. https://doi.org/10.1038/nbt.4060. - MinION nanopore sequencing to sequence human genome. Closed 12 gaps, fully typed MHC region. PCR-free sequencing preserves epigenetic modifications. Canu genome assembler. GraphMap, Minimap2 for mapping long reads. SVTyper, LUMPY for structural variants.
https://www.genengnews.com/gen-exclusives/first-nanopore-sequencing-of-human-genome/77901044

Longer reads - more errors

- The increased read length and error rate of single-molecule sequencing has challenged genome assembly programs originally designed for shorter, highly accurate reads
- Several new approaches have been developed to address this, roughly categorized as hybrid, hierarchical, or direct
  - **Hybrid methods** use single-molecule reads to reconstruct the long-range structure of the genome, but rely on complementary short reads for accurate base calls
  - **Hierarchical methods** do not require a secondary technology and instead use multiple rounds of read overlapping (alignment) and correction to improve the quality of the single-mol- ecule reads prior to assembly
  - **Direct methods** attempt to assemble single-molecule reads from a single overlapping step without any prior correction
- Hierarchical strategy is the most suitable to produce continuous *de novo* assembly

Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation

- Overlapping and assembly algorithm
- MinHash alignment process to overlap noisy sequencing reads
- Adaptive k-mer weighting to probabilistically handle repeats
- A modification of the greedy "best overlap graph" that avoids collapsing diverged repeats and haplotypes.

Koren, Sergey, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, Nicholas H. Bergman, and Adam M. Phillippy. "Canu: Scalable and Accurate Long-Read Assembly via Adaptive k-Mer Weighting and Repeat Separation." Genome Research 27, no. 5 (May 2017): 722–36. https://doi.org/10.1101/gr.215087.116.

https://github.com/marbl/canu, https://canu.readthedocs.io/en/latest/