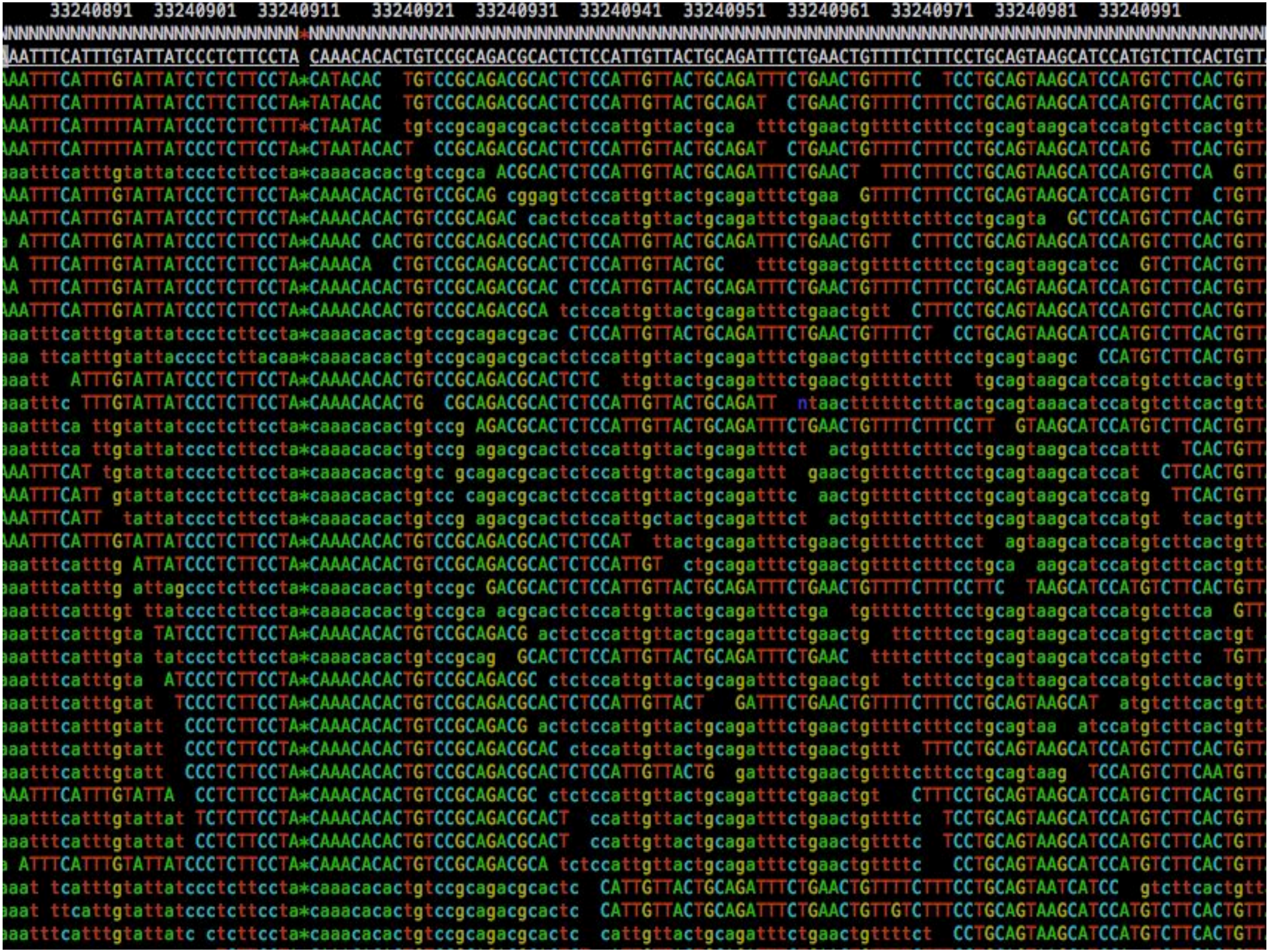


```
     33240891  33240901  33240911   33240921  33240931  33240941  33240951  33240961  33240971  33240981  33240991
NNNNNNNNNNNNNNNNNNNNNNNNNNNNN*NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
AAATTTCATTTGTATTATCCCTCTTCCTA CAAACACACTGTCCGCAGACGCACTCTCCATTGTTACTGCAGATTTCTGAACTGTTTTCTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
AAATTTCATTTGTATTATCTCTCTTCCTA*CATACAC  TGTCCGCAGACGCACTCTCCATTGTTACTGCAGATTTCTGAACTGTTTTC  TCCTGCAGTAAGCATCCATGTCTTCACTGTT
AAATTTCATTTTTATTATCCTTCTTCCTA*TATACAC  TGTCCGCAGACGCACTCTCCATTGTTACTGCAGAT  CTGAACTGTTTTCTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
AAATTTCATTTTTATTATCCCTCTTCTTT*CTAATAC  tgtccgcagacgcactctccattgttactgca  tttctgaactgttttctttcctgcagtaagcatccatgtcttcactgtt
AAATTTCATTTTTATTATCCCTCTTCCTA*CTAATACACT  CCGCAGACGCACTCTCCATTGTTACTGCAGAT  CTGAACTGTTTTCTTTCCTGCAGTAAGCATCCATG  TTCACTGTT
aaatttcatttgtattatccctcttccta*caaacacactgtccgca ACGCACTCTCCATTGTTACTGCAGATTTCTGAACT  TTTCTTTCCTGCAGTAAGCATCCATGTCTTCA  GTT
AAATTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAG cggagtctccattgttactgcagatttctgaa  GTTTTCTTTCCTGCAGTAAGCATCCATGTCTT  CTGTT
AAATTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGAC cactctccattgttactgcagatttctgaactgttttctttcctgcagta  GCTCCATGTCTTCACTGTT
a ATTTCATTTGTATTATCCCTCTTCCTA*CAAAC CACTGTCCGCAGACGCACTCTCCATTGTTACTGCAGATTTCTGAACTGTT  CTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
AA TTTCATTTGTATTATCCCTCTTCCTA*CAAACA  CTGTCCGCAGACGCACTCTCCATTGTTACTGC   tttctgaactgttttctttcctgcagtaagcatcc  GTCTTCACTGTT
AA TTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCAC CTCCATTGTTACTGCAGATTTCTGAACTGTTTTCTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
AAATTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCA tctccattgttactgcagatttctgaactgtt  CTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
aaatttcatttgtattatccctcttccta*caaacacactgtccgcagacgcac CTCCATTGTTACTGCAGATTTCTGAACTGTTTTCT  CCTGCAGTAAGCATCCATGTCTTCACTGTT
aaa ttcatttgtattacccctcttacaa*caaacacactgtccgcagacgcactctccattgttactgcagatttctgaactgttttctttcctgcagtaagc  CCATGTCTTCACTGTT
aaatt  ATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCACTCTC  ttgttactgcagatttctgaactgttttcttt  tgcagtaagcatccatgtcttcactgtt
aaatttc TTTGTATTATCCCTCTTCCTA*CAAACACACTG  CGCAGACGCACTCTCCATTGTTACTGCAGATT   ntaactttttctttactgcagtaaacatccatgtcttcactgtt
aaatttca ttgtattatccctcttccta*caaacacactgtccg AGACGCACTCTCCATTGTTACTGCAGATTTCTGAACTGTTTTCTTTCCTT  GTAAGCATCCATGTCTTCACTGTT
aaatttca ttgtattatccctcttccta*caaacacactgtccg agacgcactctccattgttactgcagatttct  actgttttctttcctgcagtaagcatccattt  TCACTGTT
AAATTTCAT tgtattatccctcttccta*caaacacactgtc gcagacgcactctccattgttactgcagattt  gaactgttttctttcctgcagtaagcatccat  CTTCACTGTT
AAATTTCATT gtattatccctcttccta*caaacacactgtcc cagacgcactctccattgttactgcagatttc  aactgttttctttcctgcagtaagcatccatg  TTCACTGTT
AAATTTCATT  tattatccctcttccta*caaacacactgtccg agacgcactctccattgctactgcagatttct  actgttttctttcctgcagtaagcatccatgt  tcactgtt
AAATTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCACTCTCCAT  ttactgcagatttctgaactgttttctttcct  agtaagcatccatgtcttcactgtt
aaatttcatttg ATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCACTCTCCATTGT  ctgcagatttctgaactgttttctttcctgca  aagcatccatgtcttcactgtt
aaatttcatttg attagccctcttccta*caaacacactgtccgc GACGCACTCTCCATTGTTACTGCAGATTTCTGAACTGTTTTCTTTCCTTC  TAAGCATCCATGTCTTCACTGTT
aaatttcatttgt ttatccctcttccta*caaacacactgtccgca acgcactctccattgttactgcagatttctga  tgttttctttcctgcagtaagcatccatgtcttca  GTT
aaatttcatttgta TATCCCTCTTCCTA*CAAACACACTGTCCGCAGACG actctccattgttactgcagatttctgaactg  ttctttcctgcagtaagcatccatgtcttcactgt
aaatttcatttgta tatccctcttccta*caaacacactgtccgcag  GCACTCTCCATTGTTACTGCAGATTTCTGAAC  ttttctttcctgcagtaagcatccatgtcttc  TGTT
aaatttcatttgta  ATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGC ctctccattgttactgcagatttctgaactgt  tctttcctgcattaagcatccatgtcttcactgtt
aaatttcatttgtat  TCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCACTCTCCATTGTTACT   GATTTCTGAACTGTTTTCTTTCCTGCAGTAAGCAT  atgtcttcactgtt
aaatttcatttgtatt  CCCTCTTCCTA*CAAACACACTGTCCGCAGACG actctccattgttactgcagatttctgaactgttttctttcctgcagtaa  atccatgtcttcactgtt
aaatttcatttgtatt  CCCTCTTCCTA*CAAACACACTGTCCGCAGACGCAC ctccattgttactgcagatttctgaactgttt  TTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
aaatttcatttgtatt  CCCTCTTCCTA*CAAACACACTGTCCGCAGACGCACTCTCCATTGTTACTG  gatttctgaactgttttctttcctgcagtaag  TCCATGTCTTCAATGTT
AAATTTCATTTGTATTA  CCTCTTCCTA*CAAACACACTGTCCGCAGACGC ctctccattgttactgcagatttctgaactgt   CTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
aaatttcatttgtattat TCTCTTCCTA*CAAACACACTGTCCGCAGACGCACT  ccattgttactgcagatttctgaactgttttc  TCCTGCAGTAAGCATCCATGTCTTCACTGTT
aaatttcatttgtattat CCTCTTCCTA*CAAACACACTGTCCGCAGACGCACT  ccattgttactgcagatttctgaactgttttc  TCCTGCAGTAAGCATCCATGTCTTCACTGTT
a ATTTCATTTGTATTATCCCTCTTCCTA*CAAACACACTGTCCGCAGACGCA tctccattgttactgcagatttctgaactgttttc   CCTGCAGTAAGCATCCATGTCTTCACTGTT
aaat tcatttgtattatccctcttccta*caaacacactgtccgcagacgcactc  CATTGTTACTGCAGATTTCTGAACTGTTTTCTTTCCTGCAGTAATCATCC  gtcttcactgtt
aaat ttcattgtattatccctcttccta*caaacacactgtccgcagacgcactc  CATTGTTACTGCAGATTTCTGAACTGTTGTCTTTCCTGCAGTAAGCATCCATGTCTTCACTGTT
aaatttcatttgtattatc ctcttccta*caaacacactgtccgcagacgcactc  cattgttactgcagatttctgaactgttttct  CCTGCAGTAAGCATCCATGTCTTCACTGTT
```

Start

Finish

GGAACTTGATGCTCAGAGAGGACAAGTCATTTGCCCAAGGTCACACAGCTGGC
AACTGGCAGACGAGATTCACGCCCTGGCAATTTGACTCCAGAATCCTAACCTT
AACCCAGAAGCACGGCTTCAAGCCCTGGAAACCACAATACCTGTGGCAGCCA
GGGGGAGGTGCTGGAATCTCATTTCACATGTGGGGAGGGGGCTCCTGTGCTC
AAGGTCACAACCAAAGAGGAAGCTGTGATTAAAACCCAGGTCCCATTTGCAAA
GCCTCGACTTTTAGCAGGTGCATCATACTGTTCCCACCCCTCCCATCCCACTTC
TGTCCAGCCGCCTAGCCCCACTTTCTTTTTTTTCTTTTTTTGAGACAGTCTCCCT
CTTGCTGAGGCTGGAGTGCAGTGGCGAGATCTCGGCTCACTGTAACCTCCGCC
TCCCGGGTTCAAGCGATTCTCCTGCCTCAGCCTCCCAAGTAGCTAGGATTACA
GGCGCCCGCCACCACGCCTGGCTAACTTTTGTATTTTTAGTAGAGATGGGGTTT
CACCATGTTGGCCAGGCTGGTCTCAAACTCCTGACCTTAAGTGATTCGCCCAC
TGTGGCCTCCCAAAGTGCTGGGATTACAGGCGTGAGCTACCGCCCCCAGCCC
CTCCCATCCCACTTCTGTCCAGCCCCCTAGCCCTACTTTCTTTCTGGGATCCAG
GAGTCCAGATCCCCAGCCCCCTCTCCAGATTACATTCATCCAGGCACAGGAAA
GGACAGGGTCAGGAAAGGAGGACTCTGGGCGGCAGCCTCCACATTCCCCTTC
CACGCTTGGCCCCCAGAATGGAGGAGGGTGTCTGTATTACTGGGCGAGGTGT
CCTCCCTTCCTGGGGACTGTGGGGGGTGGTCAAAAGACCTCTATGCCCCACCT
CCTTCCTCCCTCTGCCCTGCTGTGCCTGGGGCAGGGGGAGAACAGCCCACCTC
GTGACTGGGCTGCCCAGCCCGCCCTATCCCTGGGGGAGGGGGCGGGACAGG
GGGAGCCCTATAATTGGACAAGTCTGGGATCCTTGAGTCCTACTCAGCCCCAG
CGGAGGTGAAGGACGTCCTTCCCCAGGAGCCGGTGAGAAGCGCAGTCGGGG
GCACGGGGATGAGCTCAGGGGCCTCTAGAAAGAGCTGGGACCCTGGGAAGC
CCTGGCCTCCAGGTAGTCTCAGGAGAGCTACTCGGGGTCGGGCTTGGGGAGA
GGAGGAGCGGGGGTGAGGCAAGCAGCAGGGGACTGGACCTGGGAAGGGCT
GGGCAGCAGAGACGACCCGACCCGCTAGAAGGTGGGGTGGGGAGAGCAGCT
GGACTGGGATGTAAGCCATAGCAGGACTCCACGAGTTGTCACTATCATTATCG
AGCACCTACTGGGTGTCCCCAGTGTCCTCAGATCTCCATAACTGGGGAGCCAG
GGGCAGCGACACGGTAGCTAGCCGTCGATTGGAGAACTTTAAAATGAGGACT
GAATTAGCTCATAAATGGAACACGGCGCTTAACTGTGAGGTTGGAGCTTAGAA
TGTGAAGGGAGAATGAGGAATGCGAGACTGGGACTGAGATGGAACCGGCGGT
GGGGAGGGGGTGGGGGGATGGAATTTGAACCCCGGGAGAGGAAGATGGAAT
TTTCTATGGAGGCCGACCTGGGGATGGGGAGATAAGAGAAGACCAGGAGGGA
GTTAAATAGGGAATGGGTTGGGGGCGGCTTGGTAAATGTGCTGGGATTAGGCT
GTTGCAGATAATGCAACAAGGCTTGGAAGGCTAACCTGGGGTGAGGCCGGGT
TGGGGGCGCTGGGGGTGGGAGGAGTCCTCACTGGCGGTTGATTGACAGTTTC
TCCTTCCCCAGACTGGCCAATCACAGGCAGGAAGATGAAGGTTCTGTGGGCTG
CGTTGCTGGTCACATTCCTGGCAGGTATGGGGGCGGGGCTTGCTCGGTTCCCC
CCGCTCCTCCCCCTCTCATCCTCACCTCAACCTCCTGGCCCCATTCAGACAGAC
CCTGGGCCCCCTCTTCTGAGGCTTCTGTGCTGCTTCCTGGCTCTGAACAGCGAT
TTGACGCTCTCTGGGCCTCGGTTTCCCCCATCCTTGAGATAGGAGTTAGAAGTT
GTTTTGTTGTTGTTGTTTGTTGTTGTTGTTTTGTTTTTTTGAGATGAAGTCTCGCT

Next-Generation Sequencing (NGS) Technologies and Data Analysis

Christopher E. Mason
TA: Paul Zumbo
Spring 2010

Course Over Four Sessions:

1. Background, sample preparation, sequencing types
2. Alignments, QC, and data processing
3. Algorithms for DNA-Seq, RNA-Seq, ChIP-Seq,
4. Metagenomics, clinical genomics, data visualization and integration with other databases

Class #1:
Background, Library Preparation, Sequencing

First get the .dmg files for Xcode development

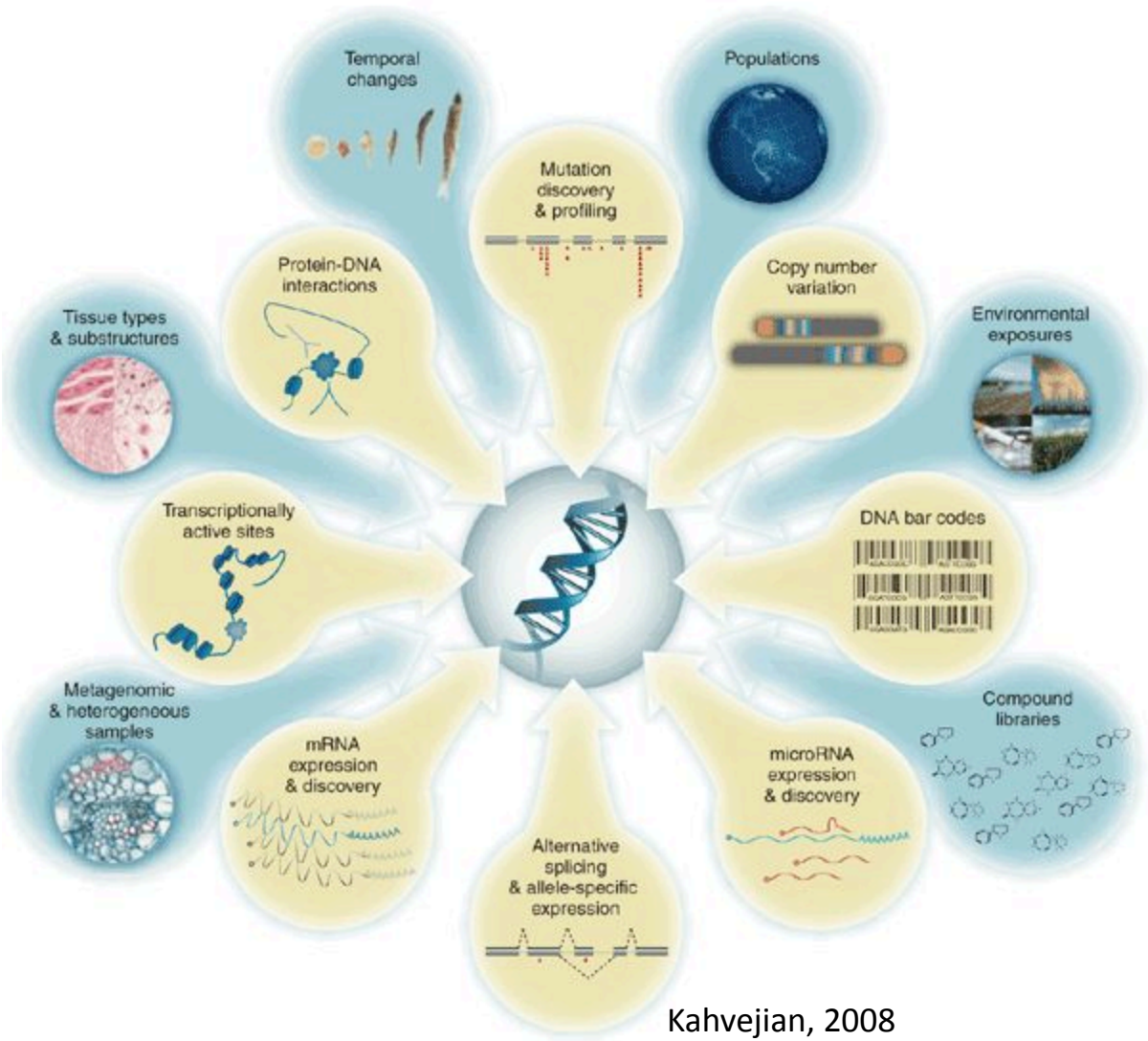
Go to the source:
http://physiology.med.cornell.edu/faculty/mason/lab/data/xcode
Xcode 3.1.4 for Leopard
Xcode 3.2.2 for Snow Leopard

These are normally found at http://connect.apple.com
Where you can sign up for a free account to develop tools on your Mac.

Also, we will download the human genome for our reference:
http://genome.ucsc.edu/
Do to Downloads → Human → Full Data Set → chromFa.tar.gz

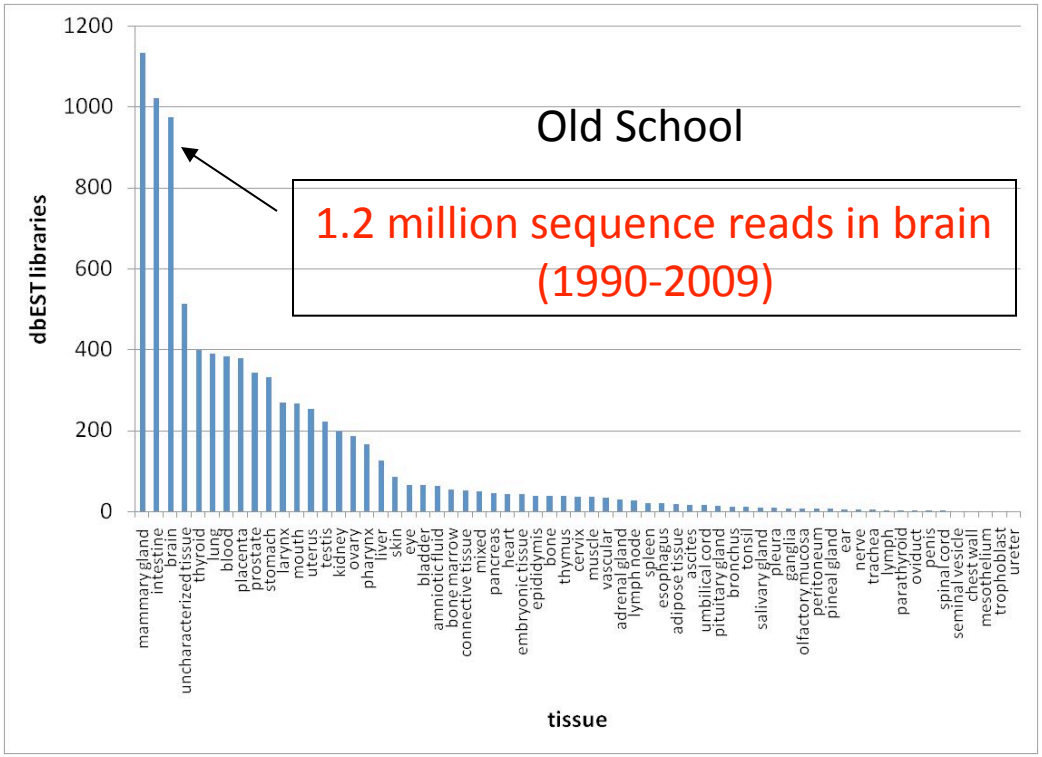
However, if you are on a cluster, many of these tools for compiling and modifying software will be installed by default on your system.

Since DNA defines the biochemical recipe for the genesis of organisms, sequencing allows us to create molecular portraits of development and disease at single-base resolution.



Kahvejian, 2008

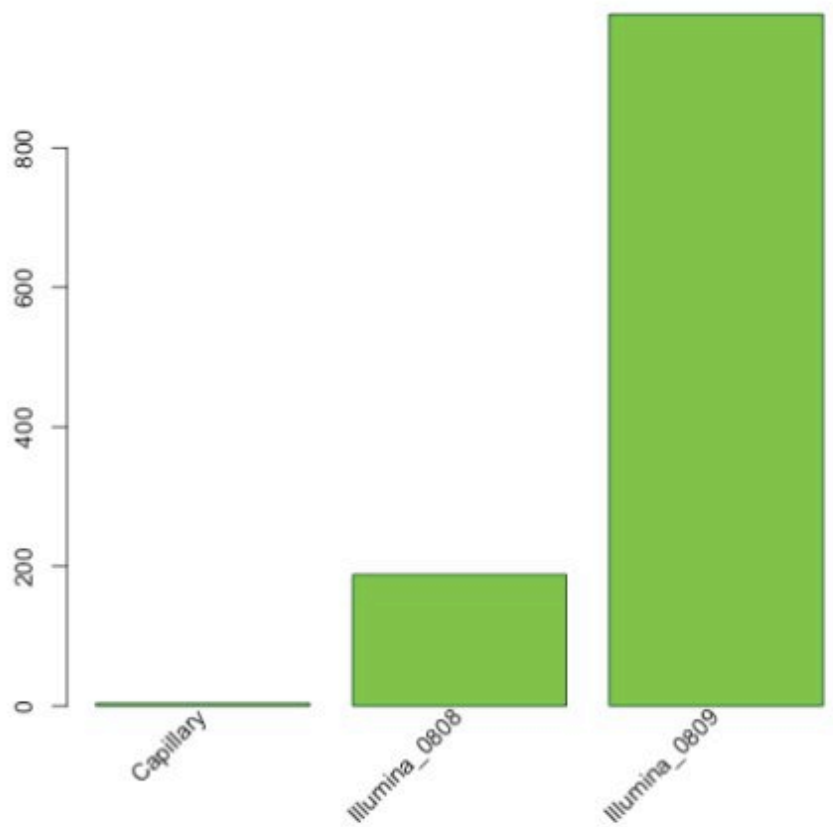
What erudition do we have now?



- Currently limited amount of EST info at NCBI
- EST data is expensive, time-consuming (cloning), and exhibits 3' bias.
- Much EST and cDNA data is for whole brains, and few libraries exist with region-specific data.

New School:
One run of a GA2 = 250 million sequence reads

A view from the Sanger



Peak world capillary monthly production: ~3.5Gb

Sanger Illumina

- August '08 production: ~188Gb
- August '09 production: ~991Gb

Estimated August 2010 production: ~20,000 Gb

Thomas Keane, 2010

But, there are many options:

Platform	Library/ template preparation	NGS chemistry	Read length (bases)	Run time (days)	Gb per run	Machine cost (US$)	Pros	Cons	Biological applications	Refs
Roche/454's GS FLX Titanium	Frag, MP/ emPCR	PS	330*	0.35	0.45	500,000	Longer reads improve mapping in repetitive regions; fast run times	High reagent cost; high error rates in homo-polymer repeats	Bacterial and insect genome *de novo* assemblies; medium scale (<3 Mb) exome capture; 16S in metagenomics	D. Muzny, pers. comm.
Illumina/ Solexa's GA_{II}	Frag, MP/ solid-phase	RTs	75 or 100	4‡, 9§	18‡, 35§	540,000	Currently the most widely used platform in the field	Low multiplexing capability of samples	Variant discovery by whole-genome resequencing or whole-exome capture; gene discovery in metagenomics	D. Muzny, pers. comm.
Life/APG's SOLiD 3	Frag, MP/ emPCR	Cleavable probe SBL	50	7‡, 14§	30‡, 50§	595,000	Two-base encoding provides inherent error correction	Long run times	Variant discovery by whole-genome resequencing or whole-exome capture; gene discovery in metagenomics	D. Muzny, pers. comm.
Polonator G.007	MP only/ emPCR	Non-cleavable probe SBL	26	5§	12§	170,000	Least expensive platform; open source to adapt alternative NGS chemistries	Users are required to maintain and quality control reagents; shortest NGS read lengths	Bacterial genome resequencing for variant discovery	J. Edwards, pers. comm.
Helicos BioSciences HeliScope	Frag, MP/ single molecule	RTs	32*	8‡	37‡	999,000	Non-bias representation of templates for genome and seq-based applications	High error rates compared with other reversible terminator chemistries	Seq-based methods	91
Pacific Biosciences (target release: 2010)	Frag only/ single molecule	Real-time	964*	N/A	N/A	N/A	Has the greatest potential for reads exceeding 1 kb	Highest error rates compared with other NGS chemistries	Full-length transcriptome sequencing; complements other resequencing efforts in discovering large structural variants and haplotype blocks	S. Turner, pers. comm.

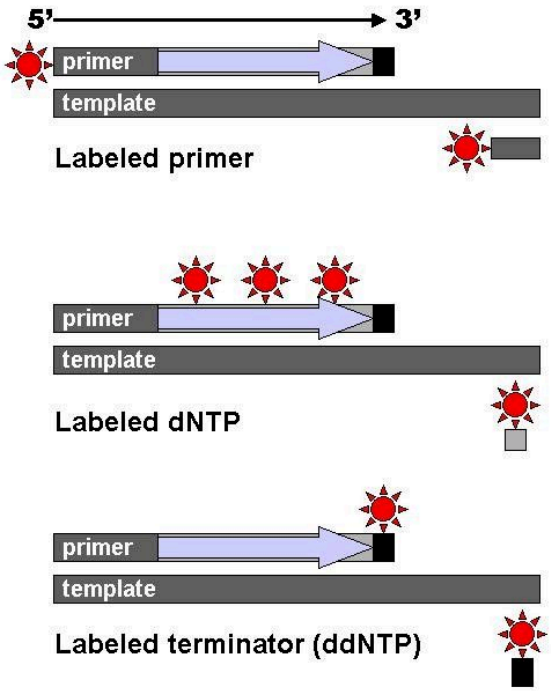
Michael Metzker, 2010

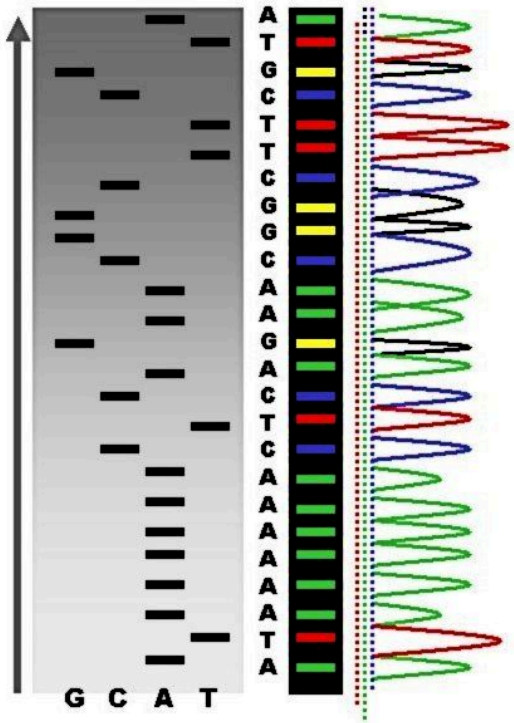
Description/Discussion of the Various Technologies

- The goal of the Archon X prize in Genomics is to enable a $1,000 genome,
- Currently at $5,000-$50,000
- Certain platforms are better suited for certain tasks:
  - Counting applications (ChIP-Seq, RNA-Seq) need more reads
  - *De novo* assembly work needs longer reads
  - Whole genome re-sequencing requires lower errors rate and high processivity

Sequencing Technologies

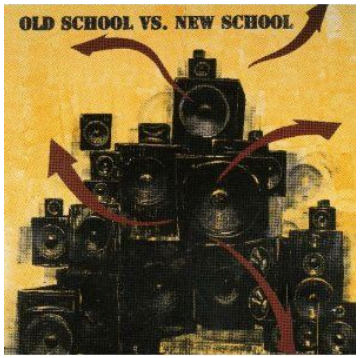
1. “Old School” dye-terminator sequencing (Sanger). 300-1000bp



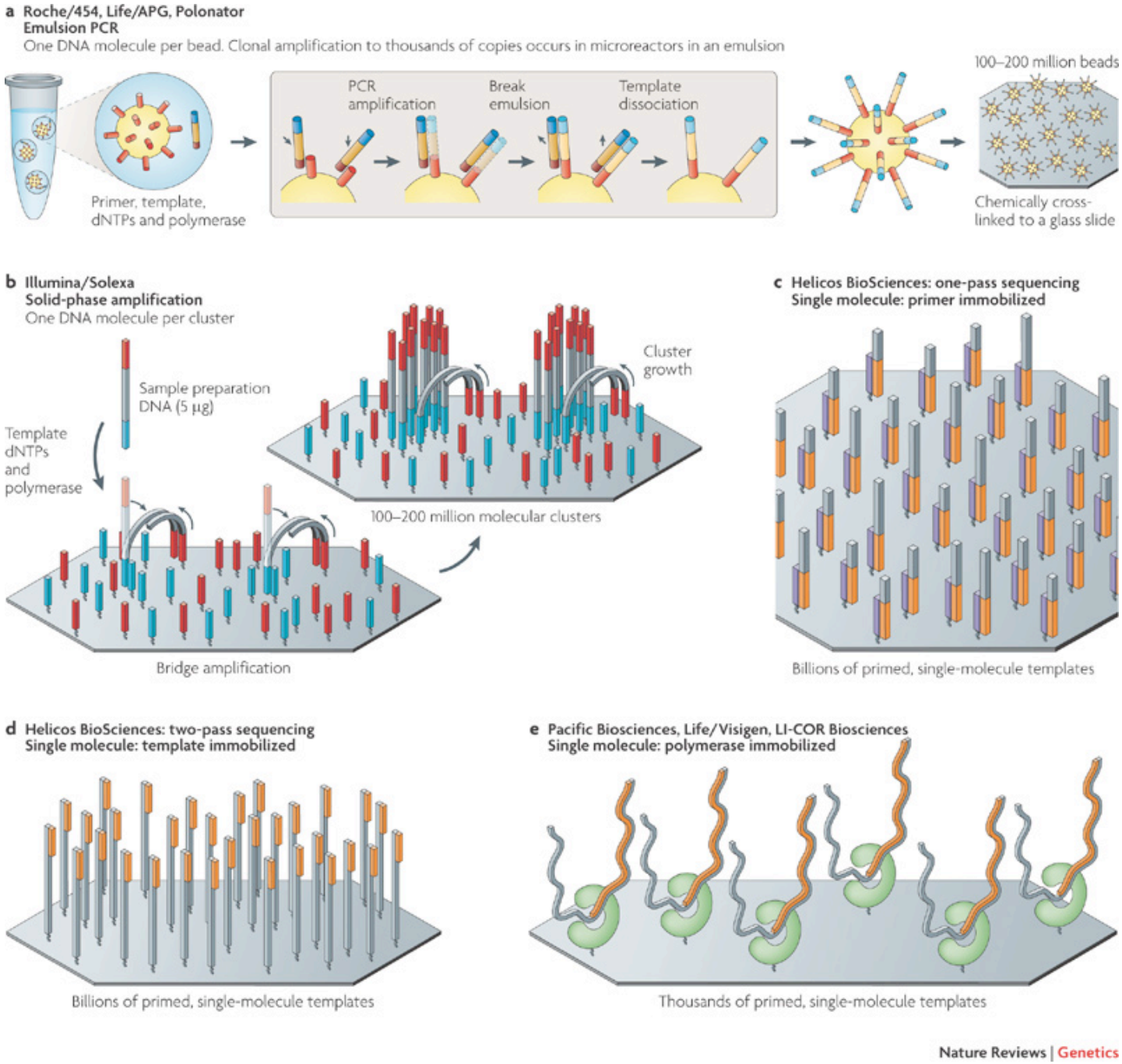


Sequencing Technologies

1. "Old School" dye-terminator sequencing (Sanger). 300-1000bp

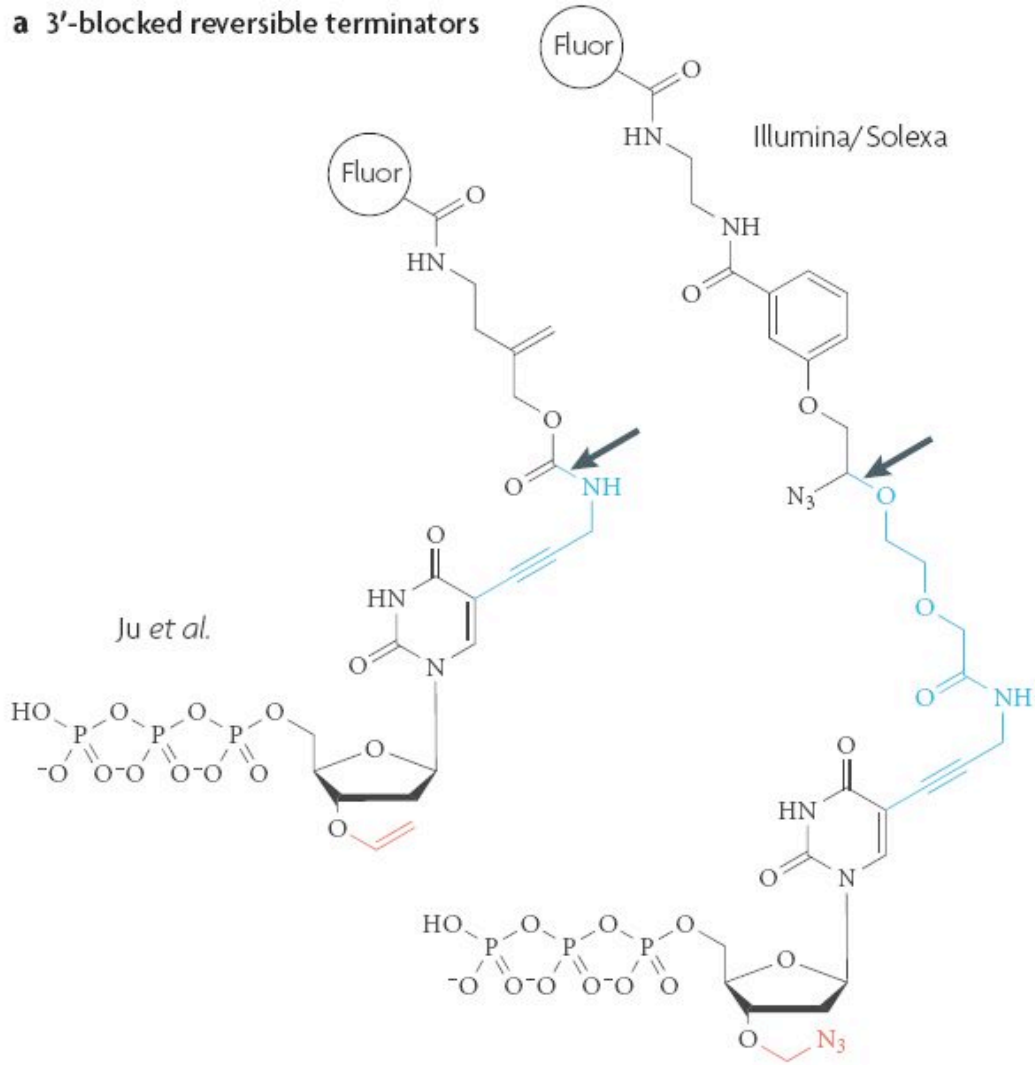


2. "New School" methods
   a. Emulsion PCR Pyrosequencing
   b. Solid-phase amplification sequencing by synthesis (clonal or single molecule)
   c. Sequencing by ligation
   d. Single-molecular, real-time (SMRT) sequencing
   e. Post-light sequencing

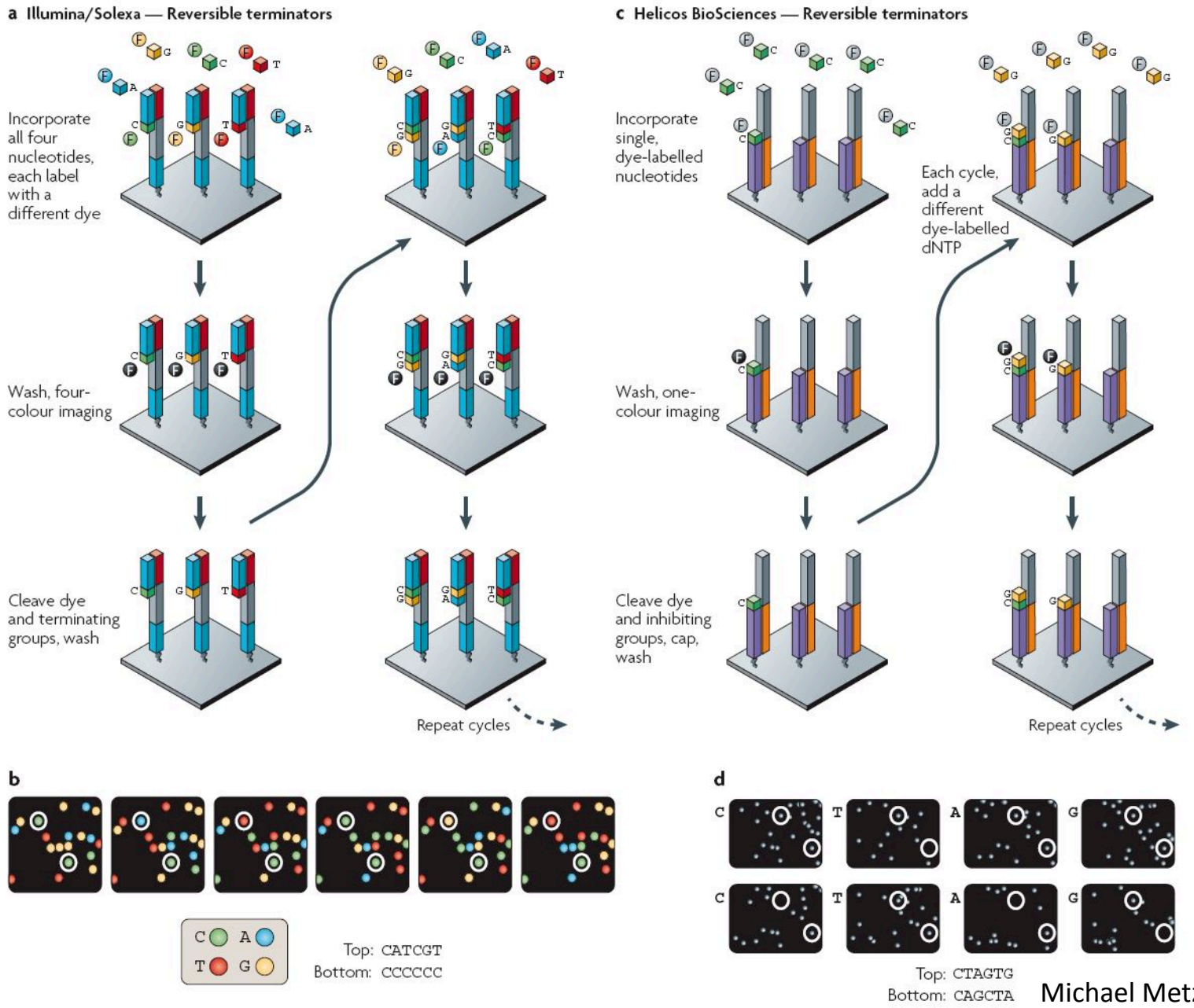


Metzker, 2010

Reversible Terminator Bases are Essential Technology Used in Most Chemistries



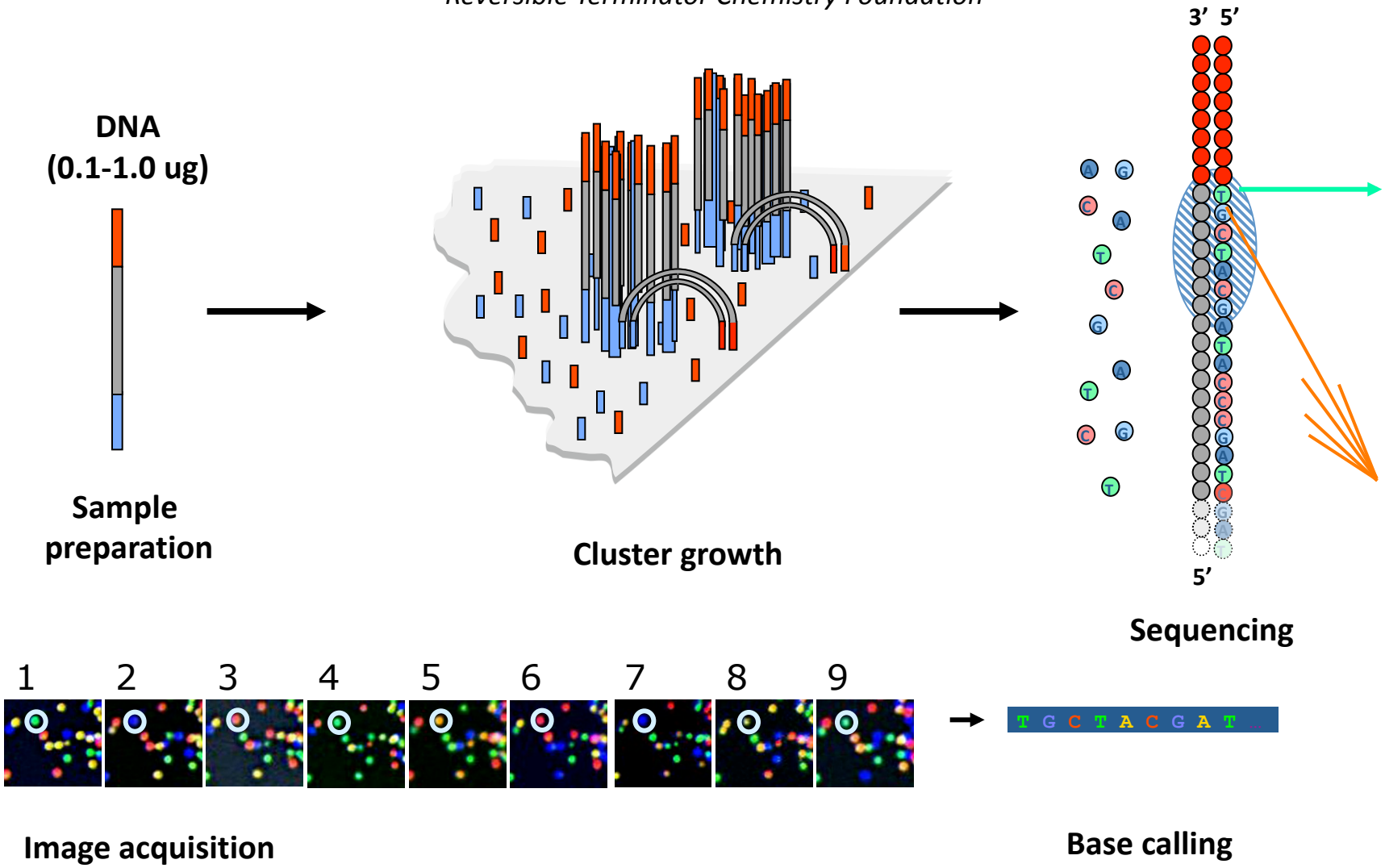
Sequencing by Synthesis (SBS)



Michael Metzker, 2010

Illumina SBS Technology

Reversible Terminator Chemistry Foundation



http://www.illumina.com/technology/sequencing_technology.ilmn

Example Workflow

1 ***Library prep (~ 6 hrs)***

Fragment DNA
↓
Repair ends / Add A overhang
↓
Ligate adapters
↓
Select ligated DNA

2 ***Automated Cluster Generation (~5 hrs)***

1-8 samples

Hybridize to flow cell
↓
Extend hybridized oligos
↓
Perform bridge amplification

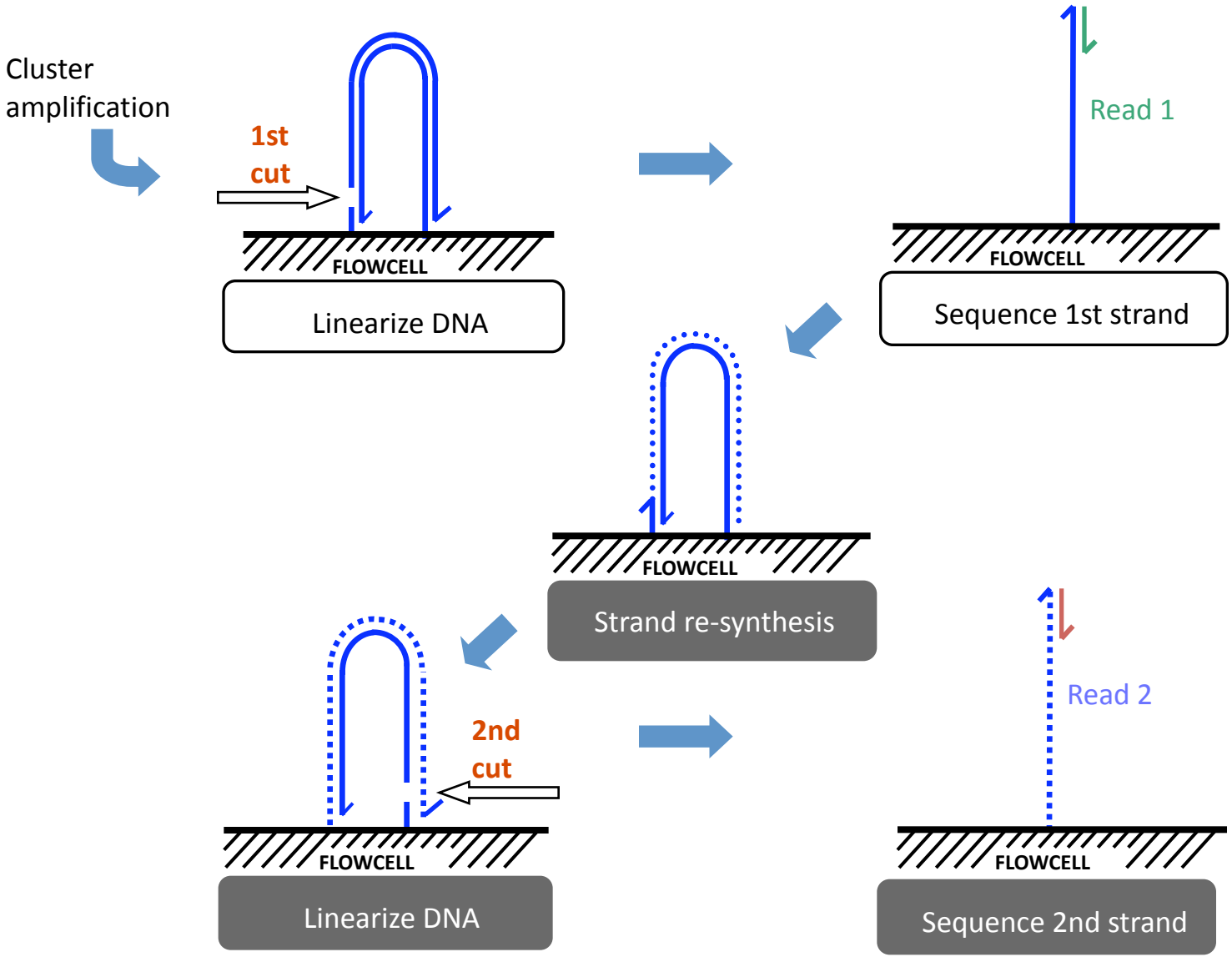
3 ***Sequencing (~ 1 to 2 days*)***

1-8 samples

Perform sequencing
↓
Generate base calls

**single read run, 18 to 36 cycles. Duration of the run depends on the desired number of sequencing cycles*

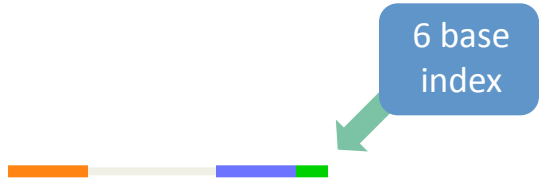
Paired-End Sequencing allows for two looks at a sequence



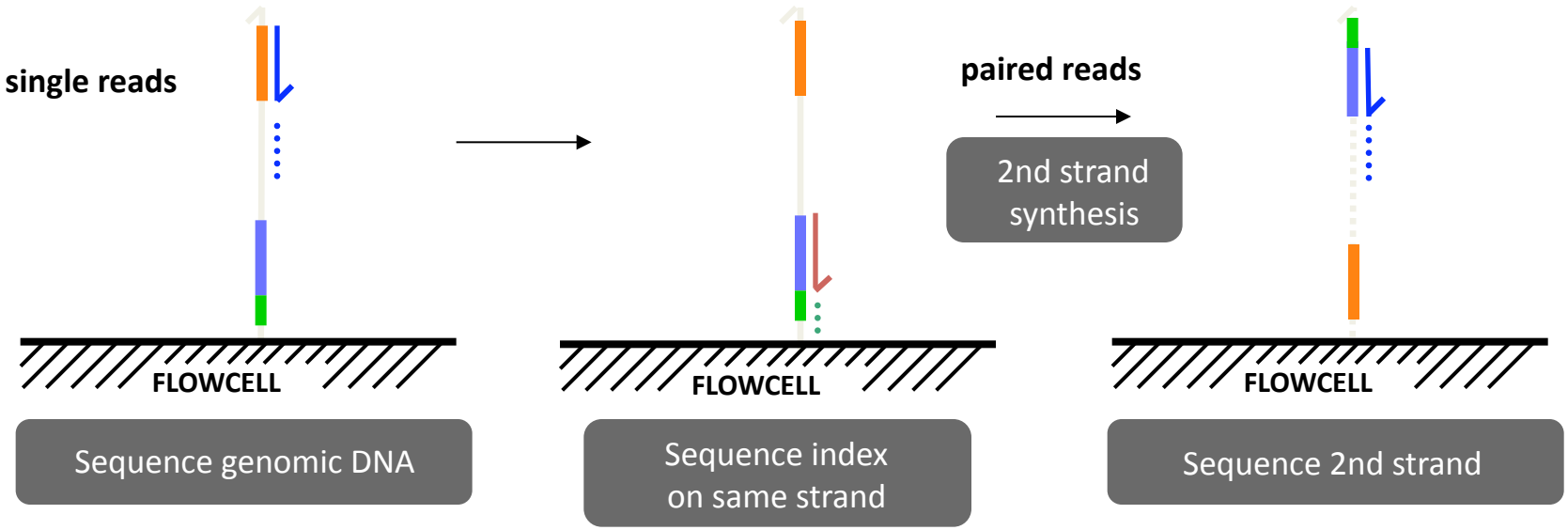
Sample Indexing

Sequence multiple samples in a single channel, reducing cost/sample

Simple construct design
Based on paired end process

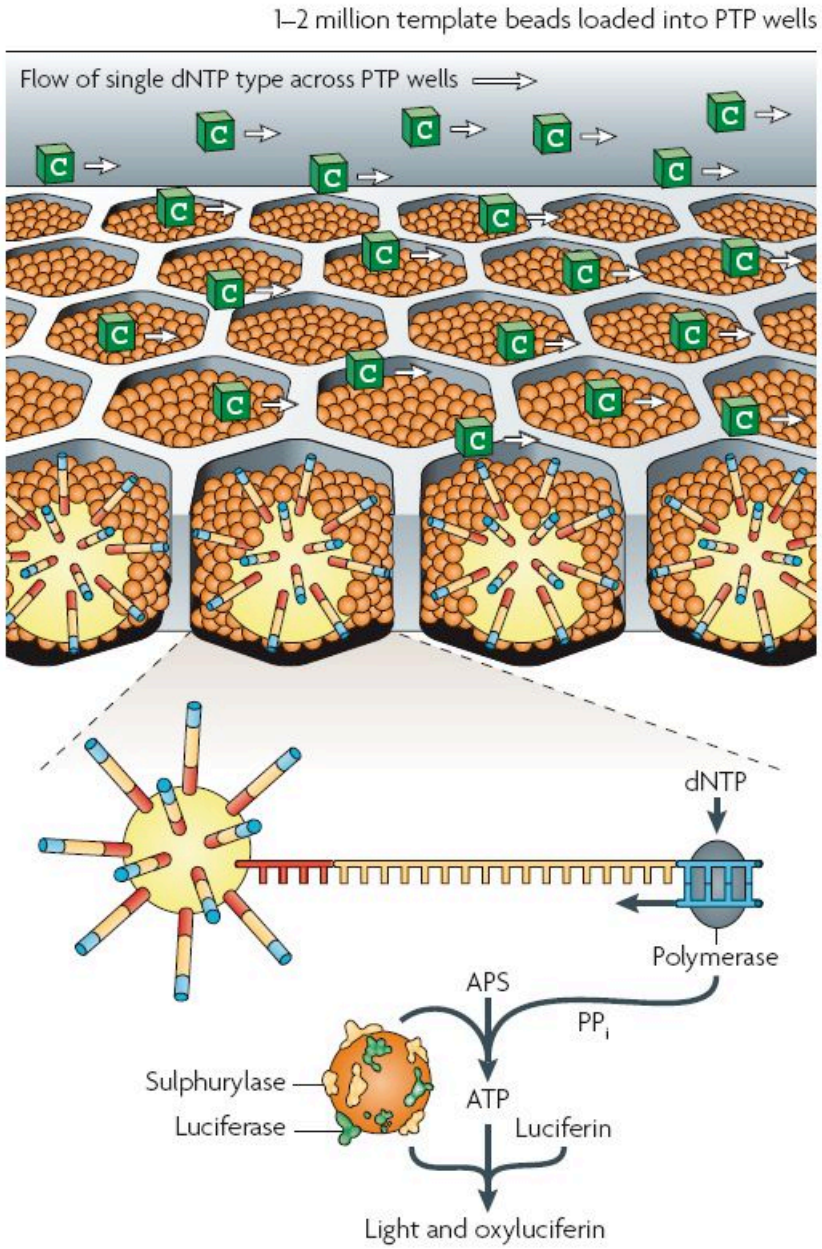


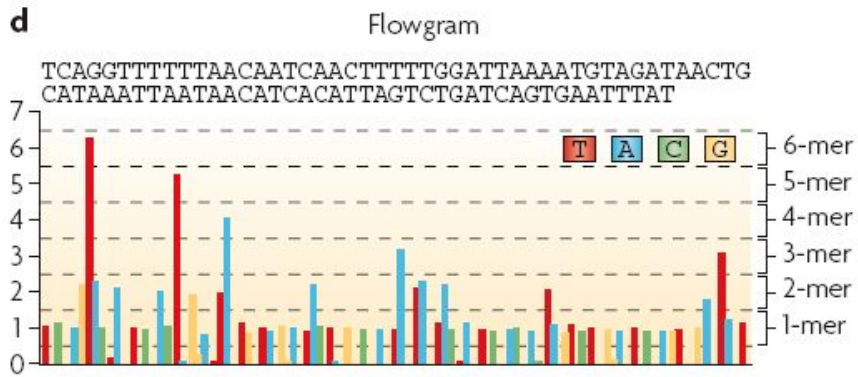
Sequencing protocol
Automated processing of indexing read

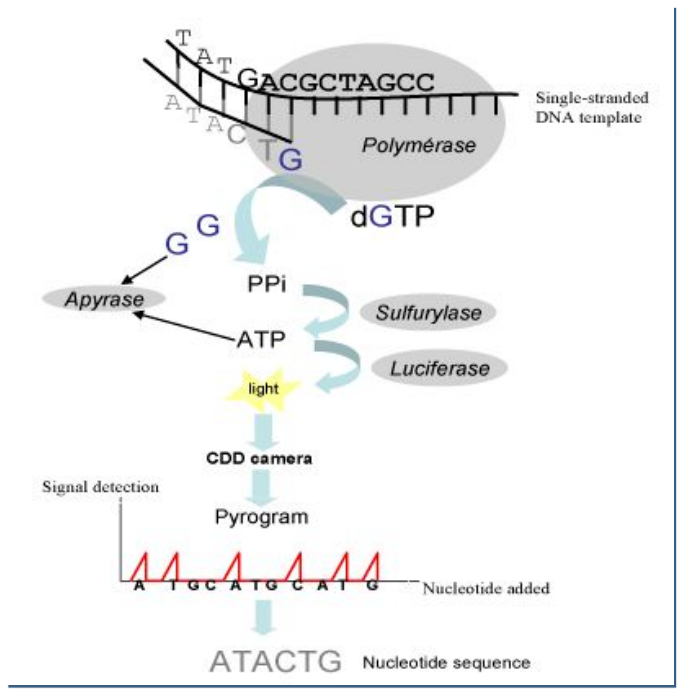


Pyrosequencing

Roche/454 — Pyrosequencing

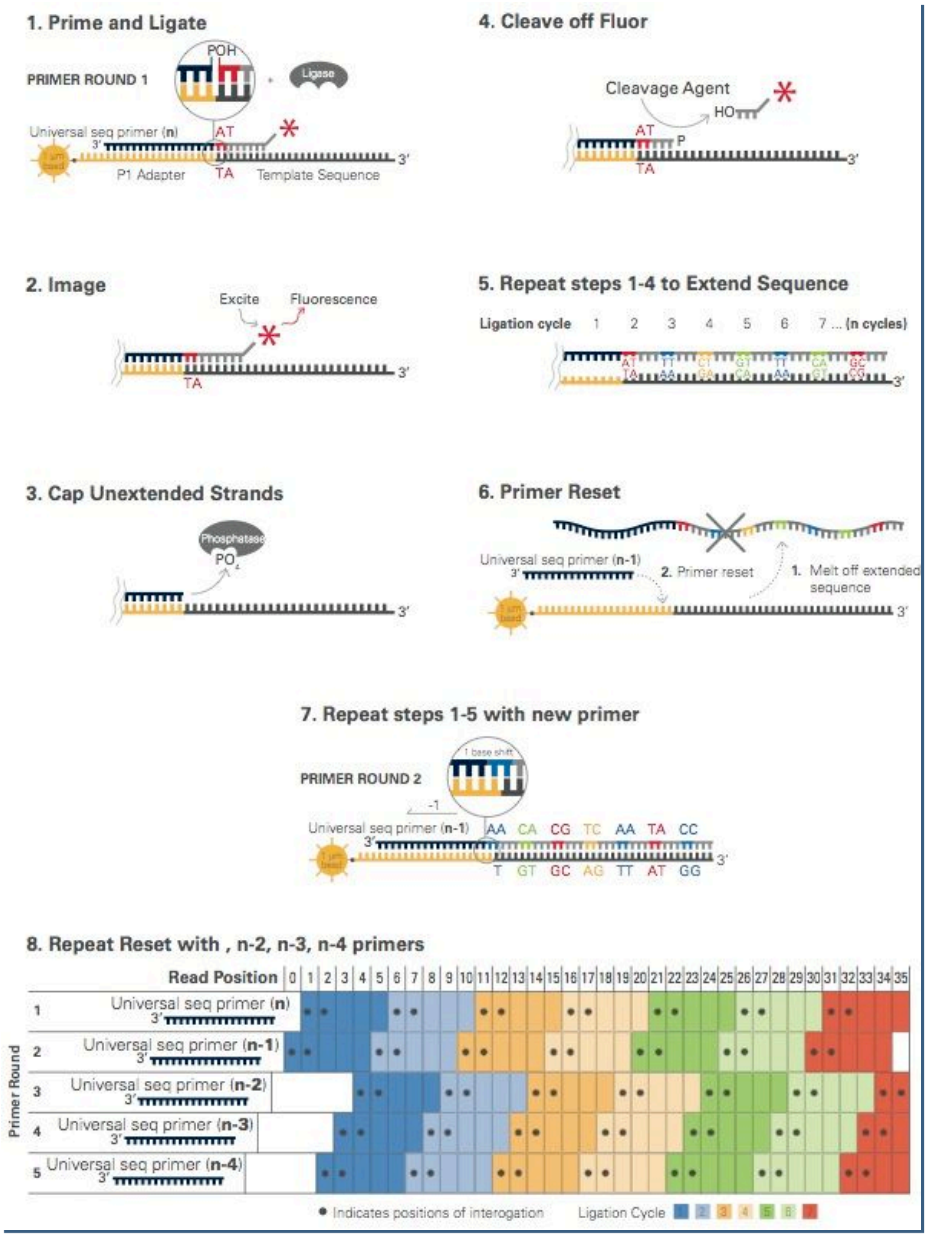




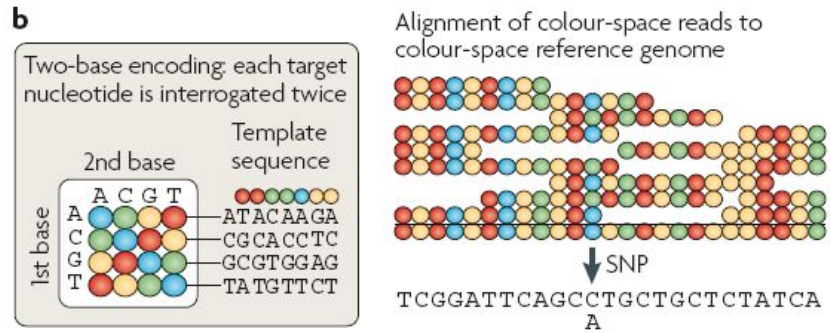


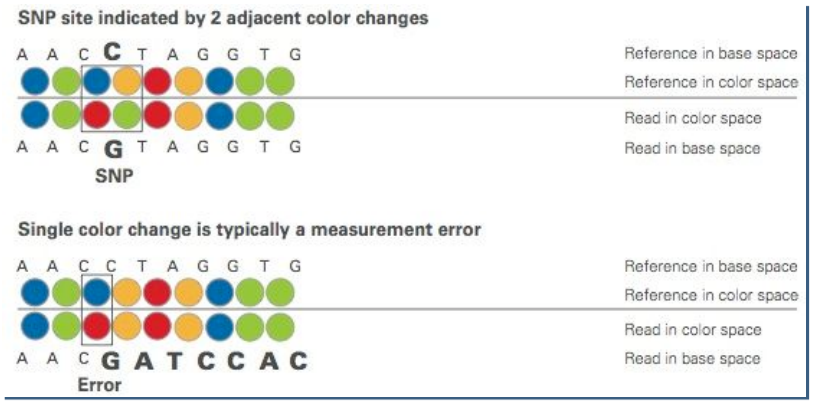
Metzker, 2010

Ligation-based Sequencing



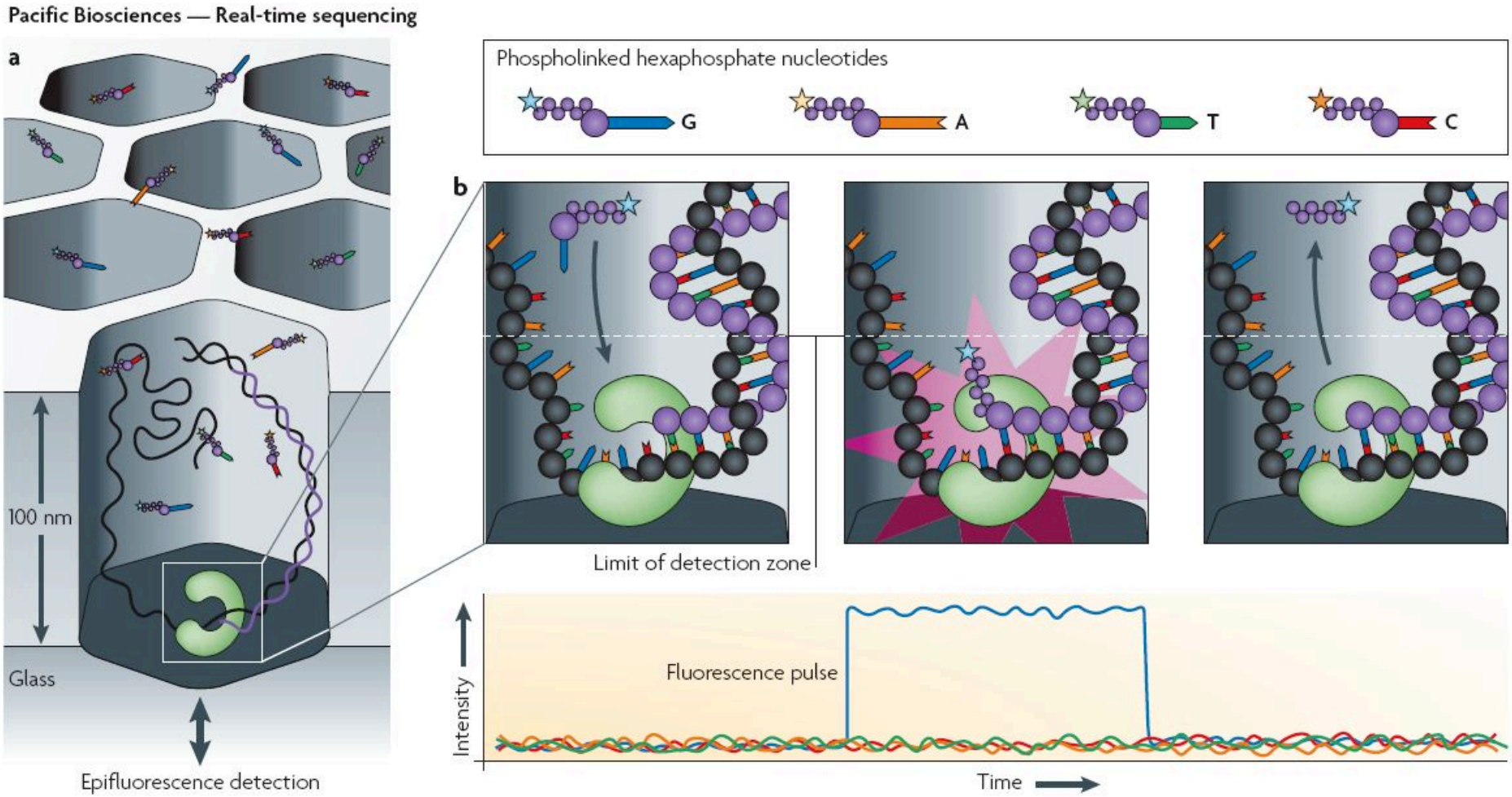
Metzker, 2010





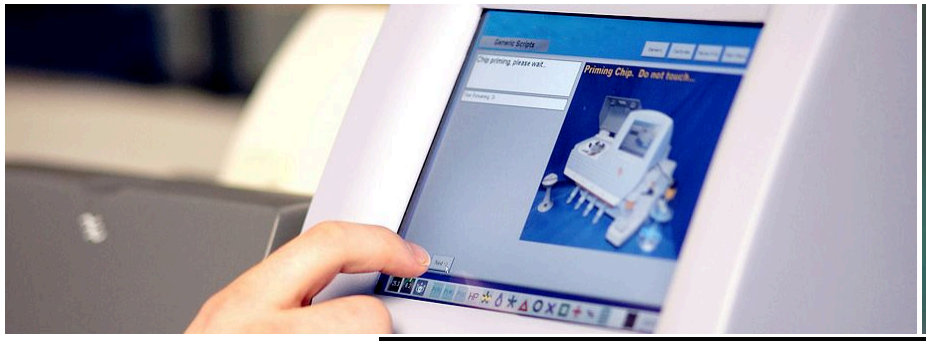
Support Oligonucleotide Ligation Detection (SOLiD)

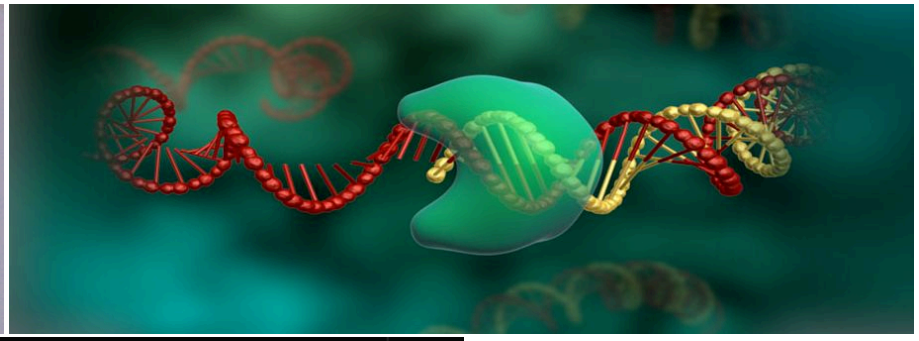
Single Molecule Real-Time

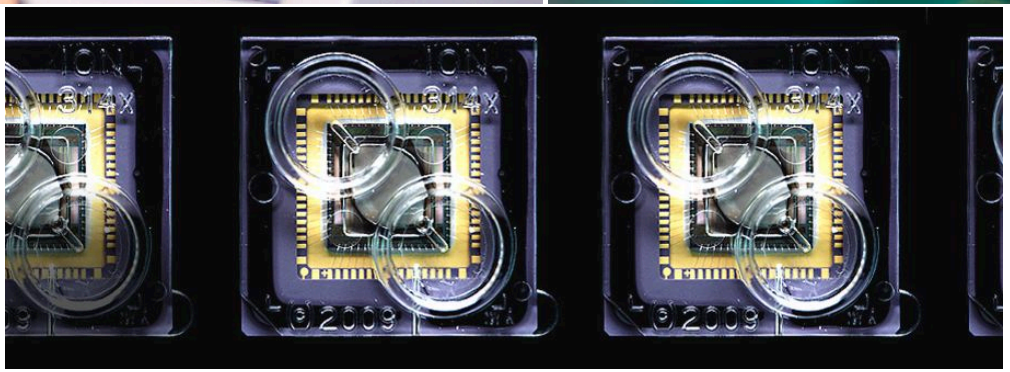


Metzker, 2010

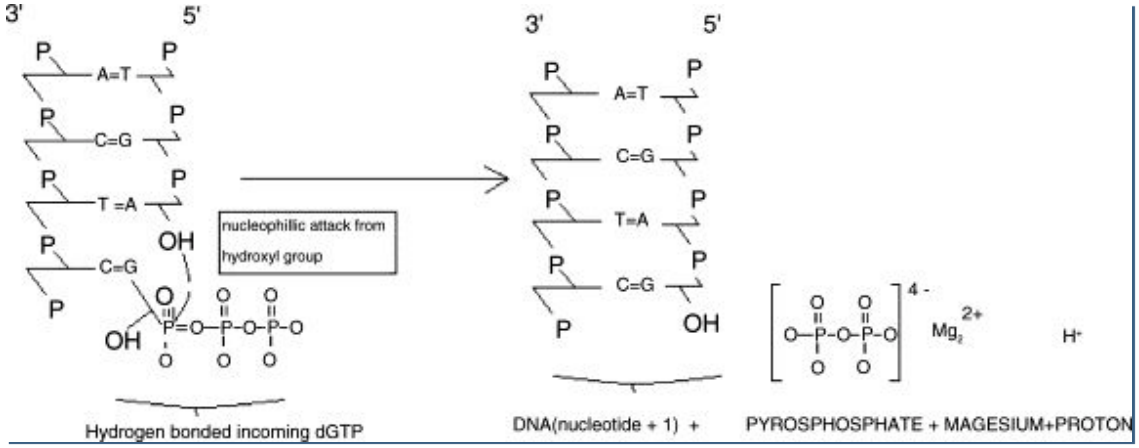
"Post-Light,"
Semi-Conductor Sequencing







Essentially,
7 million
very small
pH meters

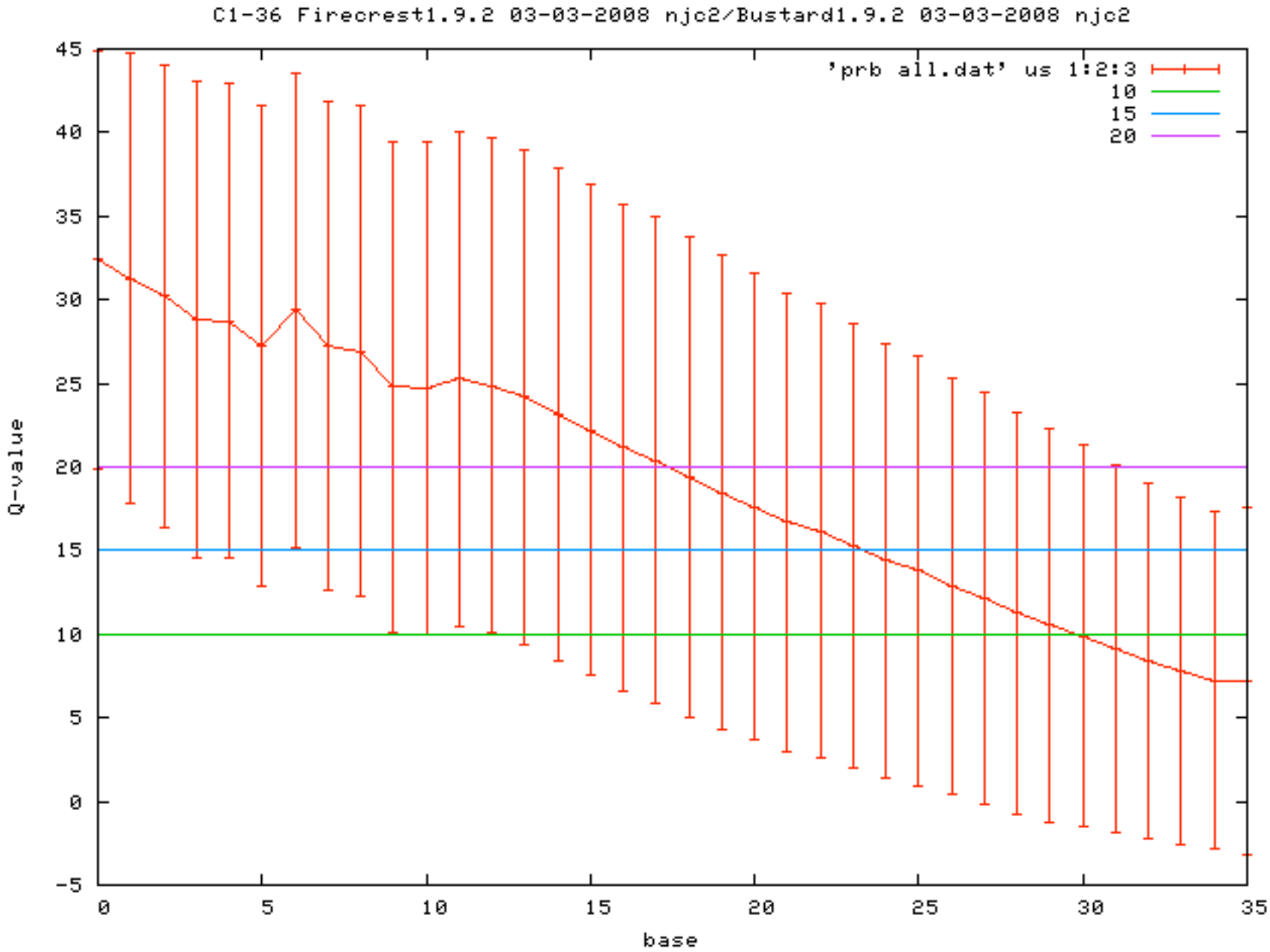


Purushothaman *et al*, 2005
IonTorrent, Inc.

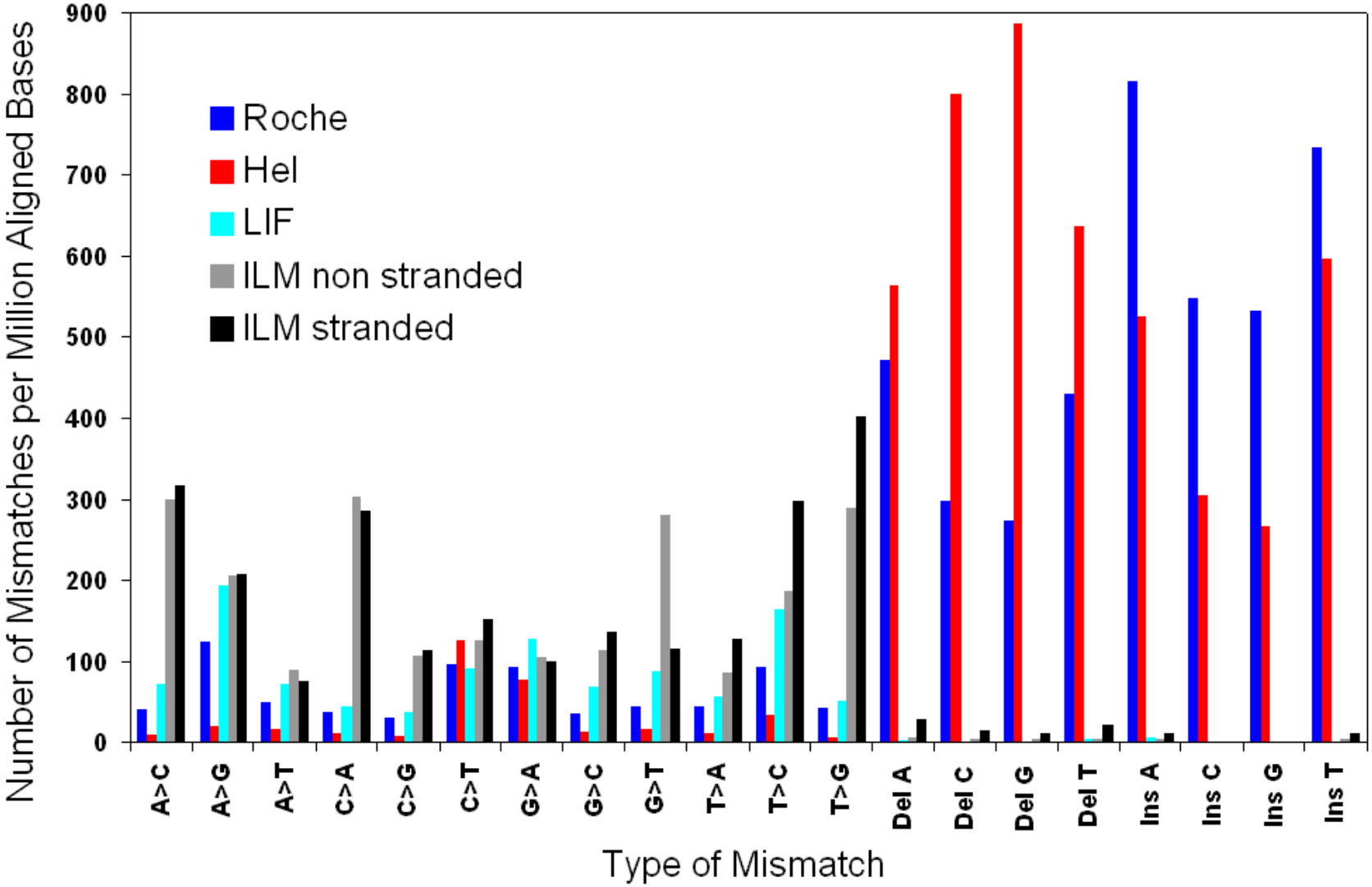
Each Platform has various sources of noise, and thus Error

- De-Phasing
  - Lagging strand dephasing from incomplete extension
  - Leading strand dephasing from over-extension
- Dark Nucleotides
- Polymerase errors (10^{-5} to 10^{-7})
- Platform-specific errors
  - Illumina more likely to have error after ‘G’
  - PCR-based methods miss GC- and AT-rich regions

Error Rates are a function of Cycle

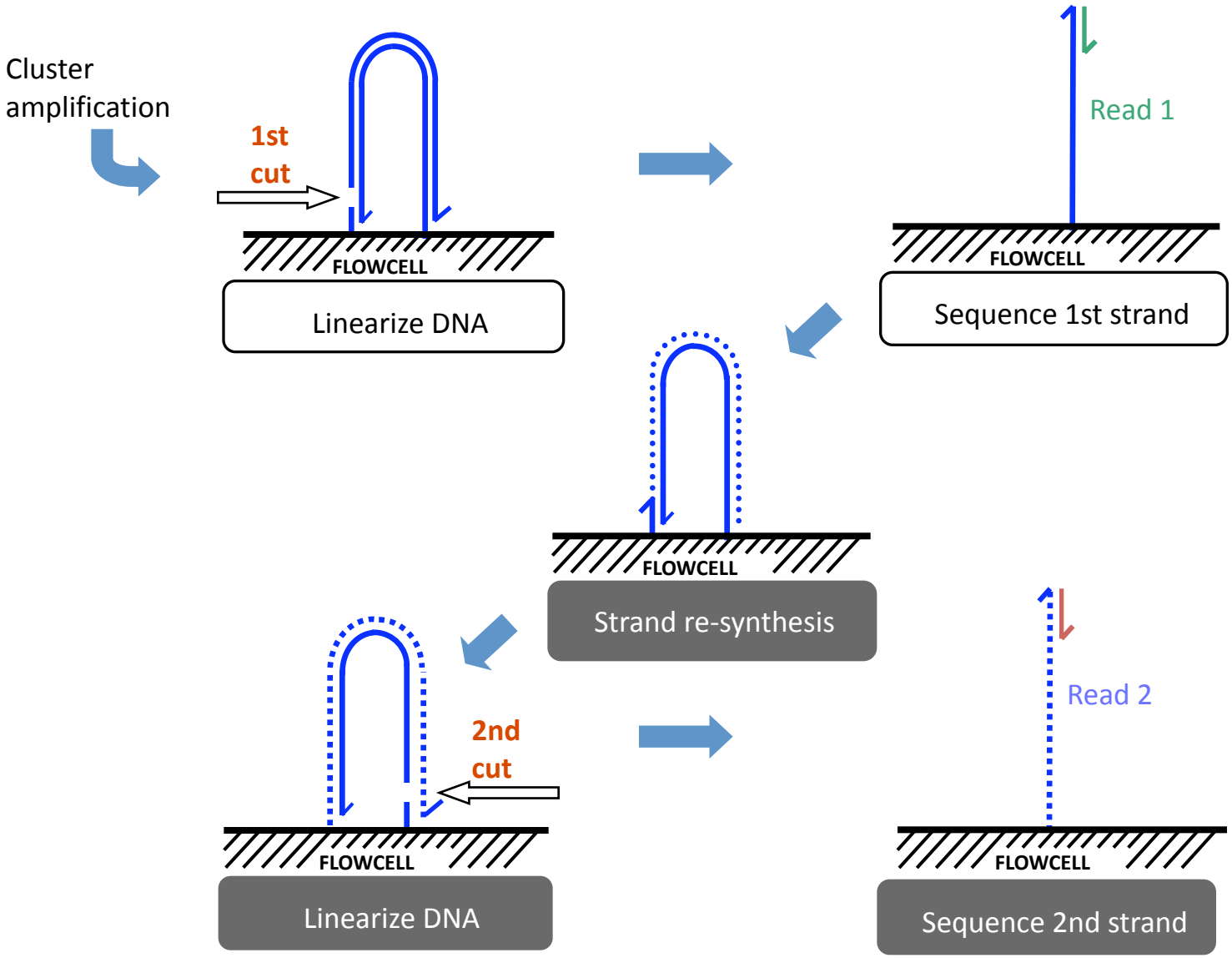


Each Platform is slightly different, and so intrinic errors are different



SeQC Consortium

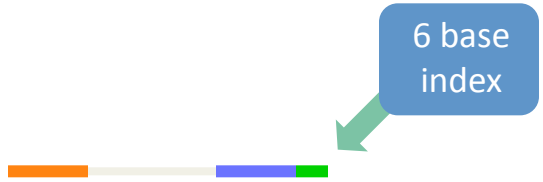
Paired-End Sequencing



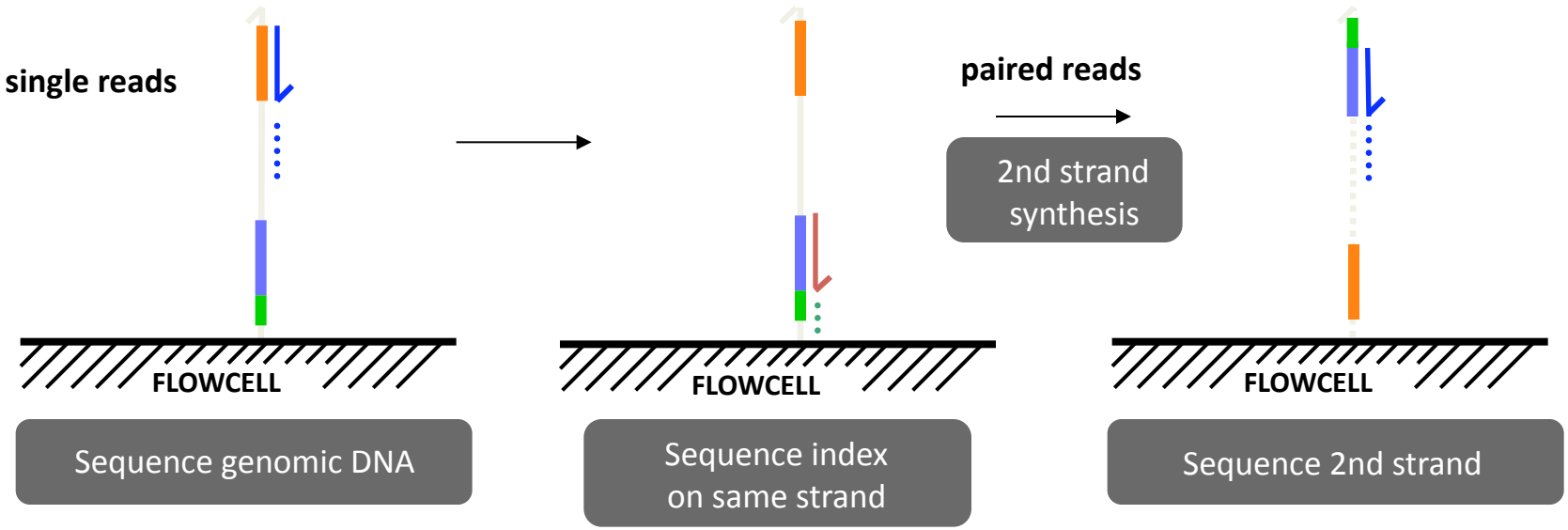
Sample Indexing

Sequence multiple samples in a single channel, reducing cost/sample

Simple construct design
Based on paired end process



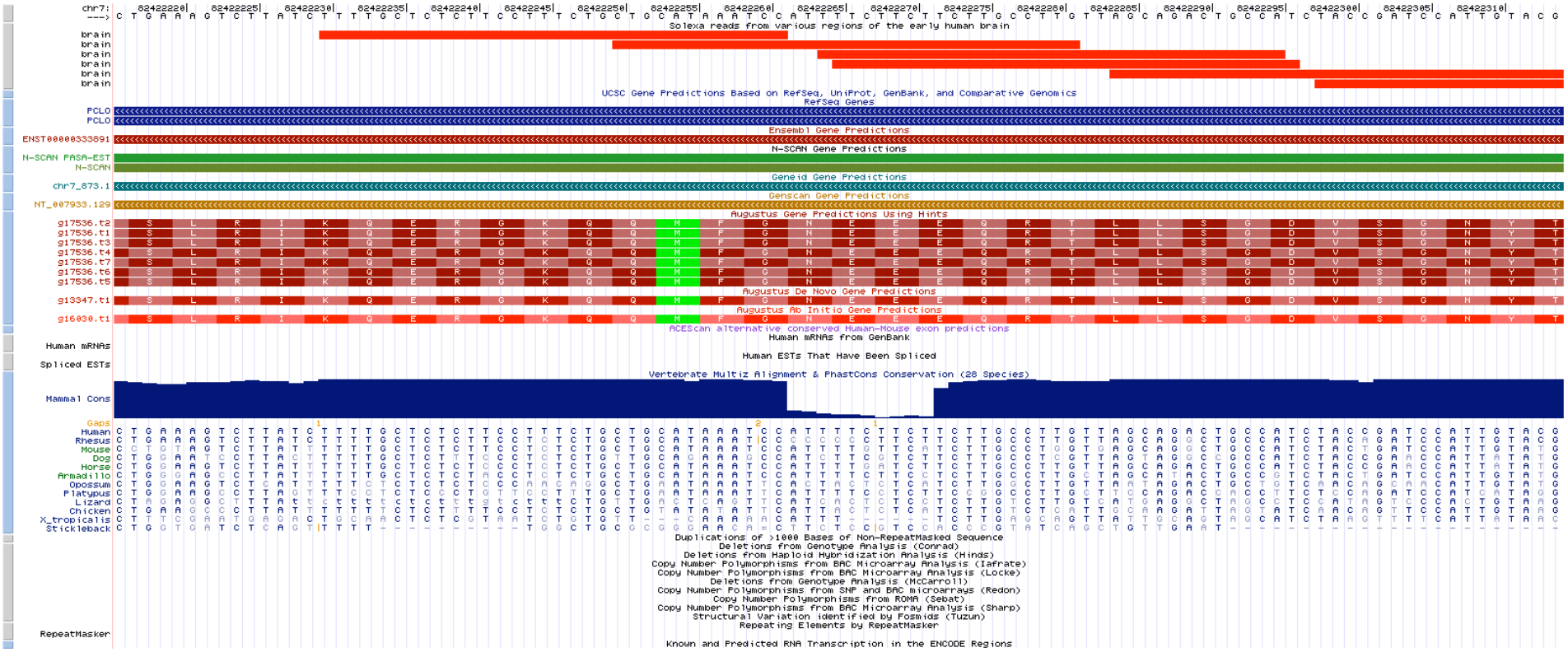
Sequencing protocol
Automated processing of indexing read

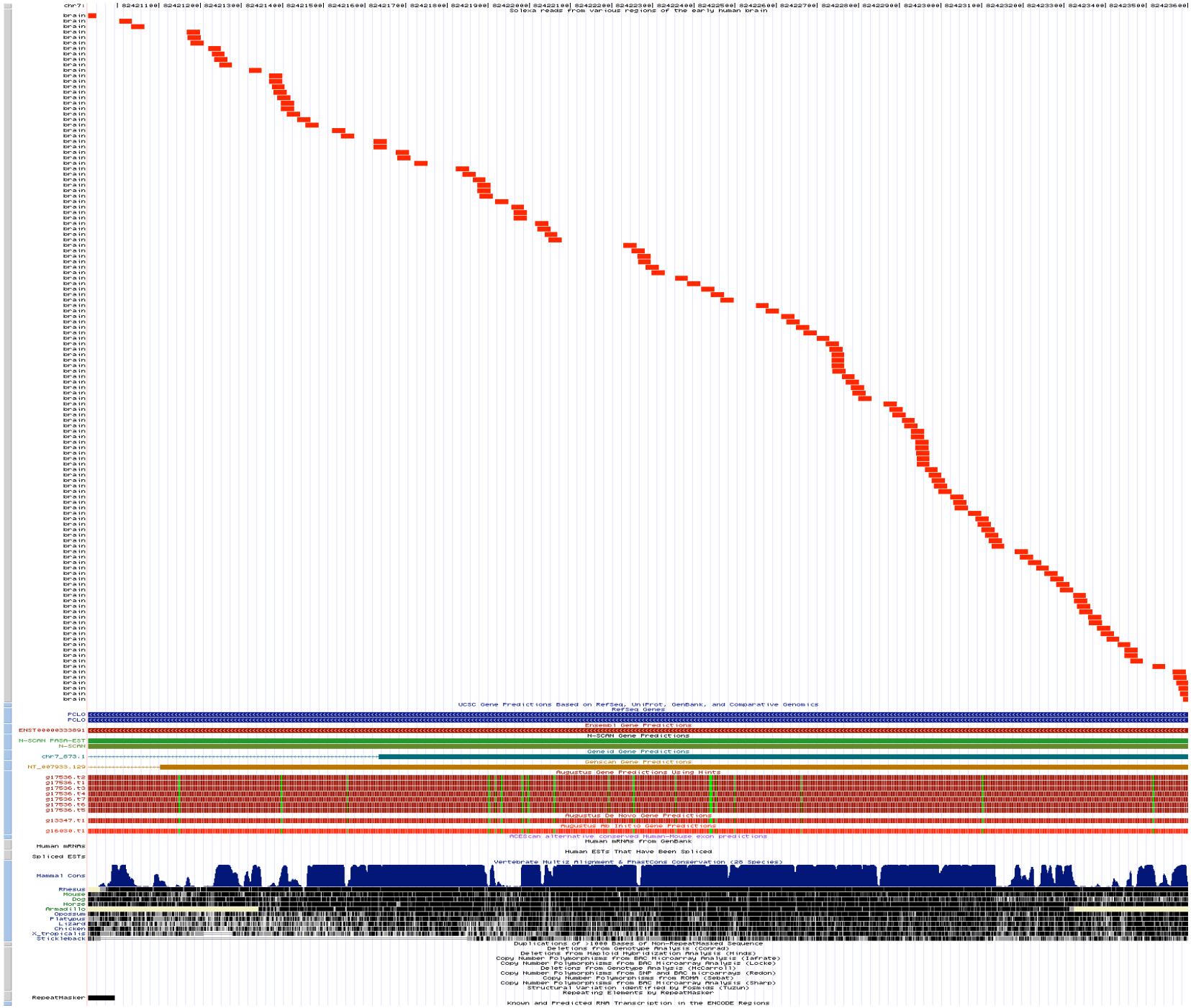


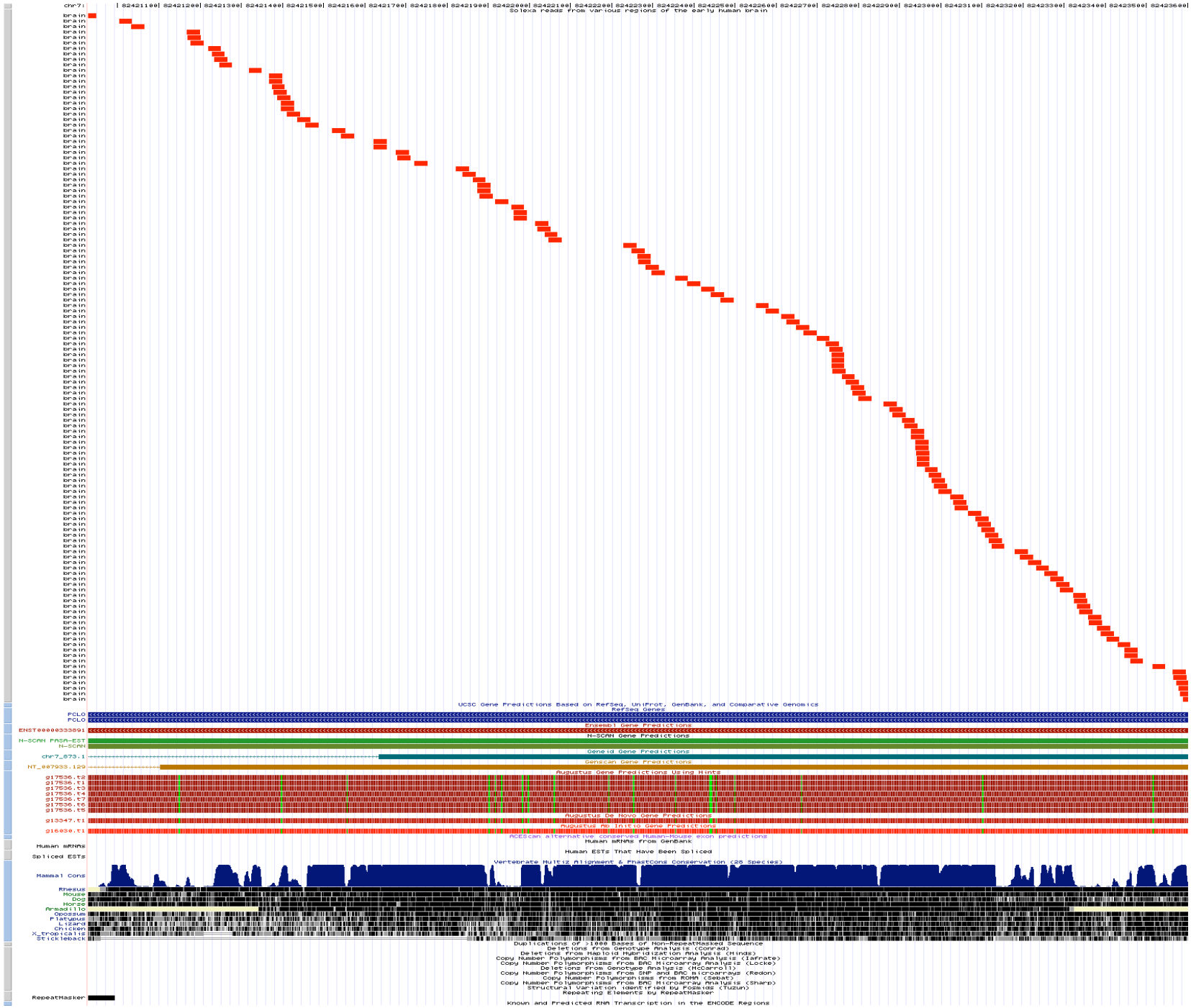
Illumina is currently the most common sequencer, so we will it for our examples.

However, we will use an aligner that has capacity to work on all systems.

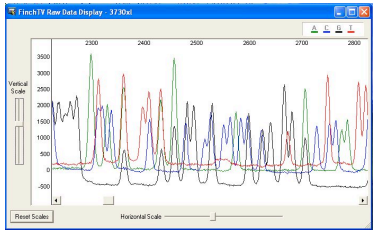
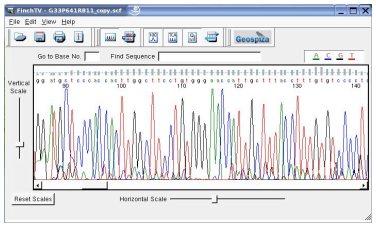
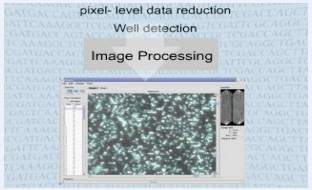
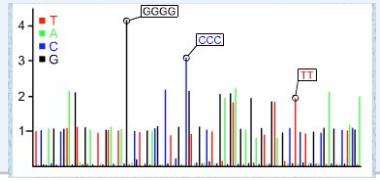
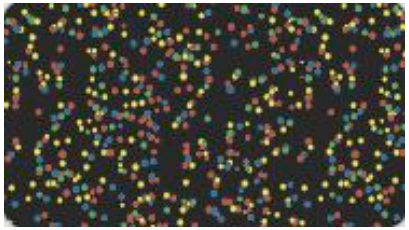
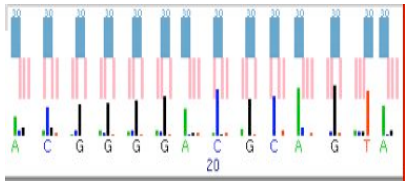
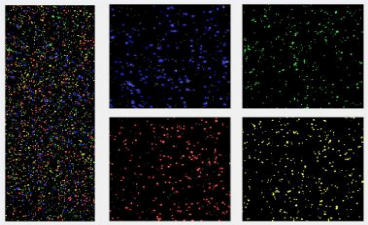
Alignment to the genome




chr7:
Solexa reads from various regions of the early human brain
UCSC Gene Predictions Based on RefSeq, UniProt, GenBank, and Comparative Genomics
RefSeq Genes
PCLO
PCLO
Ensembl Gene Predictions
ENST00000333891
N-SCAN Gene Predictions
N-SCAN PASA-EST
N-SCAN
Genid Gene Predictions
chr7_073.1
Genscan Gene Predictions
NT_007933.129
Augustus Gene Predictions Using Hints
g17536.t2
g17536.t1
g17536.t3
g17536.t4
g17536.t7
g17536.t6
g17536.t5
Augustus De Novo Gene Predictions
g13347.t1
Augustus Ab Initio Gene Predictions
g16030.t1
ACEScan alternative conserved Human-Mouse exon predictions
Human mRNAs
Human mRNAs from GenBank
Spliced ESTs
Human ESTs That Have Been Spliced
Vertebrate Multiz Alignment & PhastCons Conservation (28 Species)
Mammal Cons
Rhesus
Mouse
Dog
Horse
Armadillo
Opossum
Platypus
Lizard
Chicken
X_tropicalis
Stickleback
Duplications of >1000 Bases of Non-RepeatMasked Sequence
Deletions from Genotype Analysis (Conrad)
Deletions from Human Genotype Analysis (Hinds)
Copy Numbers Polymorphisms from Array Analysis (Iafrate)
Copy Number Polymorphisms from BAC Microarray Analysis (Locke)
Deletions from Genotype Analysis (McCarroll)
Copy Number Polymorphisms from SNP and BAC microarrays (Redon)
Copy Number Polymorphisms from ROMA (Sebat)
Copy Number Polymorphisms from BAC Microarray Analysis (Sharp)
Structural Variation identified by Fosmids (Tuzun)
Repeating Elements by RepeatMasker
RepeatMasker
Known and Predicted RNA Transcription in the ENCODE Regions


chr7:
Solexa reads from various regions of the early human brain
UCSC Gene Predictions Based on RefSeq, UniProt, GenBank, and Comparative Genomics
RefSeq Genes
PCLO
PCLO
Ensembl Gene Predictions
ENST00000333891
N-SCAN Gene Predictions
N-SCAN PASA-EST
N-SCAN
Genid Gene Predictions
chr7_073.1
Genscan Gene Predictions
NT_007933.129
Augustus Gene Predictions Using Hints
g17536.t2
g17536.t1
g17536.t3
g17536.t4
g17536.t7
g17536.t6
g17536.t5
Augustus De Novo Gene Predictions
g13347.t1
Augustus Ab Initio Gene Predictions
g16030.t1
ACEScan alternative conserved Human-Mouse exon predictions
Human mRNAs
Human mRNAs from GenBank
Spliced ESTs
Human ESTs That Have Been Spliced
Vertebrate Multiz Alignment & PhastCons Conservation (28 Species)
Mammal Cons
Rhesus
Mouse
Dog
Horse
Armadillo
Opossum
Platypus
Lizard
Chicken
X_tropicalis
Stickleback
Duplications of >1000 Bases of Non-RepeatMasked Sequence
Deletions from Genotype Analysis (Conrad)
Deletions from Human Genotype Analysis (Hinds)
Copy Numbers Polymorphisms from Array Analysis (Iafrate)
Copy Number Polymorphisms from BAC Microarray Analysis (Locke)
Deletions from Genotype Analysis (McCarroll)
Copy Number Polymorphisms from SNP and BAC microarrays (Redon)
Copy Number Polymorphisms from ROMA (Sebat)
Copy Number Polymorphisms from BAC Microarray Analysis (Sharp)
Structural Variation identified by Fosmids (Tuzun)
Repeating Elements by RepeatMasker
RepeatMasker
Known and Predicted RNA Transcription in the ENCODE Regions

Data Types

Technology	Raw Data	Primary Data	Data Formats
CE (Sanger) Sequencing by Synthesis Chain termination, electrophoretic separation with fluorescent detection			Binary files; one / sample AB1, SCF common ESD, ZTR, RCF, SRF, fasta, quality values, others
454 Sequencing by SynthesisPyrosequenicng, multiple rounds of light detection			Binary and text files; one group / plate section, "flow space," SFF, fasta, quality values, SRF
Illumina GA Sequencing by Synthesis Reversible terminators, multiple rounds of fluorescent detection			Binary and text files; one group / plate section, fasta, fastq, quality values, SRF
SOLiD Sequencing by Ligation Dibase encoding, multiple rounds of extension and fluorescent detection		CSFasta >Myseq1 G012310223…	Binary and text files; one group / plate section, "color space," csfasta, quality values, SRF

from Todd Smith,
@ Geospiza, Inc

Install Xcode

Many Options for Alignment - 2009

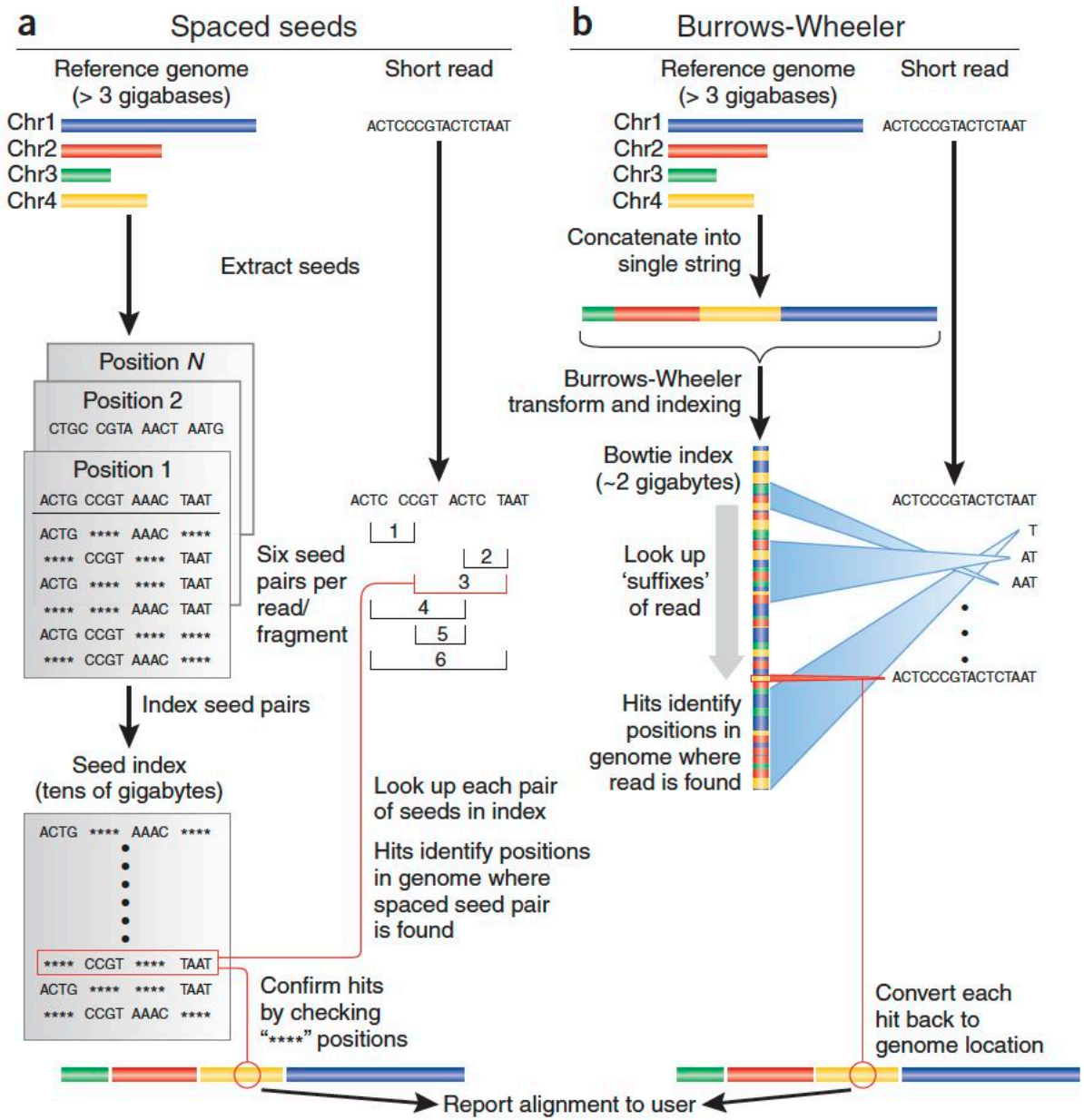
	MAQ	ELAND	SOAP	BFAST	Bowtie	SHRiMP	Rmap	SeqMap	Novocraft
Algorithm Parameters									
Version	0.71	1.1	1.11	0.1.11	0.9.8	1.1.0	0.41	1.0.8	1.06
SNP-calls	✓	-	✓	-	-	✓	-	-	-
Uses Quality Scores	✓	-	-	✓	✓	✓	✓	-	✓
Indels	PE only	PE only	✓	✓	-	✓	-	✓	-
Splicing	-	-	-	-	-	-	-	-	-
Paired-End	✓	✓	✓	✓	-	-	-	-	✓
Threading	-	✓	✓	✓	✓	-	-	-	✓
Max # Mismatches (*in Seed)	3*	2*	5	-	3*, or UD	-	-	2	7
Default Seed Size	10	32	-	-	28	-	-	-	-
Max Input Length	63	-	60	-		-	64	-	-
5' Read Trimming	-	✓	-	-	✓	-	-	-	-
3' Read Trimming	✓	✓	✓	-	✓	-	-	-	✓
Methylation Alignment	-	-	-	✓	-	-	-	-	-
Repeats/Adaptor Removal	✓	✓	-	✓	✓	-	-	-	✓
Strand-specific search	-	-	✓	-	-	-	-	✓	-
Platforms									
ABI SOLiD	✓		✓	✓	✓	✓			
Illumina GA	✓	✓	✓	✓	✓	✓	✓	✓	✓
Roche 454					✓	✓			
Helicos Heliscope		✓	✓					✓	

Many Options for Alignment - 2010

- Bfast
- BioScope
- Bowtie
- BWA
- CLC bio
- CloudBurst
- Eland/Eland2
- GenomeMapper
- GnuMap
- Karma
- MAQ
- MOM
- Mosaik
- MrFAST/MrsFAST
- NovoAlign
- PASS
- PerM
- RazerS
- RMAP
- SSAHA2
- Segemehl
- SeqMap
- SHRiMP
- Slider/SliderII
- SOAP/SOAP2
- Srprism
- Stampy
- vmatch
- ZOOM
- .....

Li et al, 2010

Many Options for Alignment - 2010



Trapnell and Salzberg, 2010

Burrows-Wheeler Transformation (BWT)

- First discovered in 1983 by Wheeler at AT&T Bell Labs
- Used for compression in 1994.
- First implemented for aligners with "Bowtie"
  Ben Langmead, Cole Trapnell, Mihai Pop,
  and Steven Salzberg
- Allows for fast searching with a small memory footprint

http://bio-bwa.sourceforge.net/

Li H. and Durbin R. "Fast and accurate short read alignment with Burrows-Wheeler transform." (2009) *Bioinformatics*, 25, 1754-60.

Burrows M, Wheeler DJ. "A Block Sorting Lossless Data Compression Algorithm." Technical Report 124. Palo Alto, CA: Digital Equipment Corporation; 1994.

Burrows-Wheeler Transformation (BWT)

```
agcagcagact
agcagcagact$
gcagcagact$a
cagcagact$ag
agcagact$agc
gcagact$agca
cagact$agcag
agact$agcagc
gact$agcagca
act$agcagcag
ct$agcagcaga
t$agcagcagac
$agcagcagact
```

	suffix#	BWT(S)	suffix/rotation	
0	11	t	$agcagcagact	$...
1	8	g	act$agcagcag	a...
2	6	c	agact$agcagc	
3	3	c	agcagact$agc	
4	0	$	agcagcagact$	
5	5	g	cagact$agcag	c...
6	2	g	cagcagact$ag	
7	9	a	ct$agcagcaga	
8	7	a	gact$agcagca	g...
9	4	a	gcagact$agca	
10	1	a	gcagcagact$a	
11	10	c	t$agcagcagac	t...

BWT(agcagcagact) = tgcc$ggaaaac

ch	$	a	c	g	t
rank(ch)	0	1	5	8	11

t$agcagcagac

Quality Scores

The most common format is FASTQ, based off the FASTA data format:

```
>SequenceID
CGTAGTCTATATATGCGCGAATGCGTA
```

But….

FASTQ also includes quality information:

```
@Sample_Info
CCTTGCTGCC
+
3.6;#$!>><
```

Understanding FASTQ

For Illumina, sequences have an ID:

@HWUSI-EAS100R:6:73:941:1973#0/1

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	'x'-coordinate of the cluster within the tile
1973	'y'-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)
/1	the member of a pair, /1 or /2 (paired-end or mate-pair reads only)

Understanding Quality Scores

Q-values are the probability (p) of a base being incorrect. From Sanger sequencing:

$Q_{value}=-10\log_{10}p$

So, if your p=0.1, then	Q_{value} =	$(-10\log_{10}(0.1))$
	=	$(-10(-1)) = 10$
If your p=0.01, then	Q_{value} =	$(-10\log_{10}(0.01))$
	=	$(-10(-2)) = 20$
If p=0.001, then	Q_{value} =	$(-10\log_{10}(0.001))$
	=	$(-10(-3)) = 30$

Understanding Quality Scores

Q-values are the probability (p) of a base being incorrect, but it is most efficient to represent this with a single bit in ASCII (American Standard Code for Information Interchange) format.

The first 32 symbols in ASCII are control characters, so we start at 33.

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 ........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                          |    |        |                              |                     |
33                         59   64       73                            104                   126

S - Sanger       Phred+33,  41 values  (0, 40)
I - Illumina 1.3 Phred+64,  41 values  (0, 40)
X - Solexa       Solexa+64, 68 values (-5, 62)
```

Phred-Based Base Quality

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.........................................................
 ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 ............................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                          |    |        |                              |                     |
33                         59   64       73                            104                   126

S - Sanger       Phred+33,  41 values  (0, 40)
I - Illumina 1.3 Phred+64,  41 values  (0, 40)
X - Solexa       Solexa+64, 68 values (-5, 62)
```

If your ASCII character is ‘B’, then 66-64=2, so

$P=10^{-Q/10}$

$-0.2 = \log_{10}p$

$10^{-0.2}$ =p, so p=0.63, or 63% change of an incorrect base.

If your ASCII character is ‘h’, then 104-64=39, so

40 $= (-10\log_{10}p)$

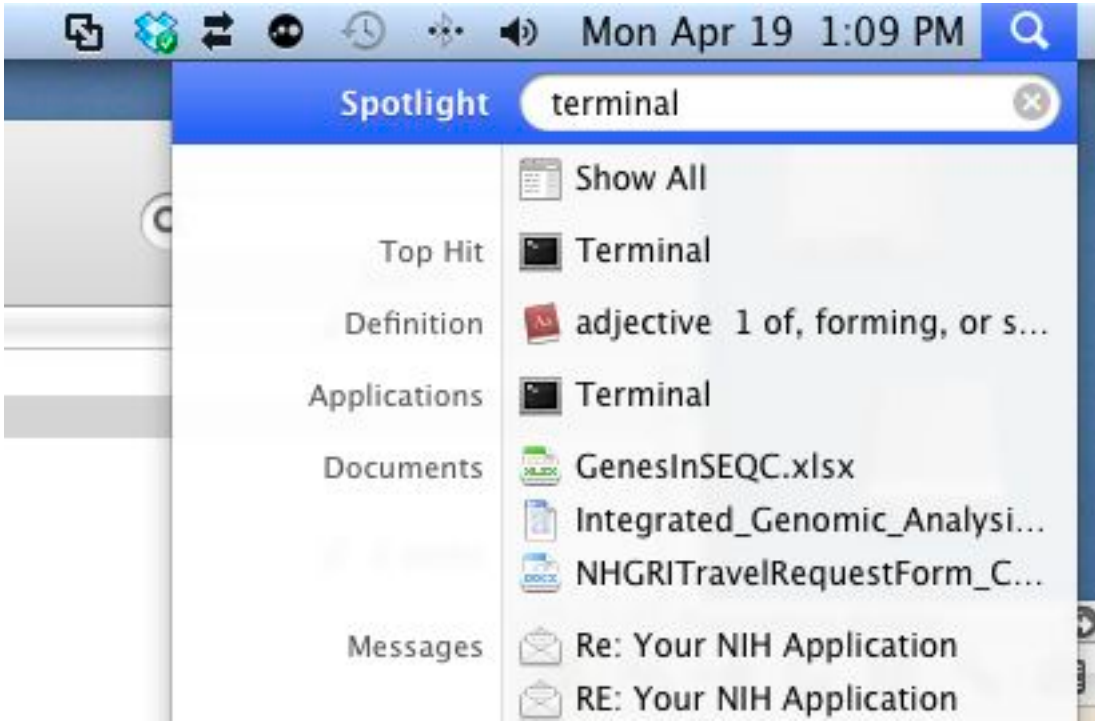
-4.0 $= \log_{10}p$

10^{-5} =p, so p=0.0001, or 0.01% change of an incorrect base.

Let’s Try it with some data

Open a Terminal

Look in Spotlight



Commonly Used UNIX commands

- **ssh**: Connect to another machine :

  ssh yourlogin@server.name,

  mkdir: `Creating a directory'

  cd: `Changing your current working directory' ('cd ' takes you home, 'cd ..' takes you up one directory)

  ls: `Finding out what files you have in current directory'

  cp: `Making a copy of a file'

  mv: `Changing the name of a file or moving a file'

  rm: `remove (delete) files': WARNING no undelete!

  chmod: `Controlling access to your files'

  cmp: Comparing two files

  wc: Word, line, and character count

  compress (or zip): Compress a file

  gzip (zip) or gunzip: (Unzip a file)

  bzip2 or bunzip2

  emacs: `Using the emacs text editor'

  man (x): gives the manual for any function here

Setting up an aligner - BWA

Go to the source:
http://sourceforge.net/projects/bio-bwa/files/

To download the file:
] http://sourceforge.net/projects/bio-bwa/files/bwa-0.5.7.tar.bz2/download

Unzip the file:
]bunzip2 bwa-0.5.7.tar.bz2

Untar the file:
]tar -xvf bwa-0.5.7.tar

cd into the new dir, then Compile the Program:
]make

Run the Program:
]./bwa

```
[cem34@bulldogi bwa-0.5.7]$ ./bwa

Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.5.7 (r1310)
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
         aln           gapped/ungapped alignment
         samse         generate alignment (single ended)
         sampe         generate alignment (paired ended)
         bwasw         BWA-SW for long queries

         fa2pac        convert FASTA to PAC format
         pac2bwt       generate BWT from PAC
         pac2bwtgen    alternative algorithm for generating BWT
         bwtupdate     update .bwt to the new format
         pac_rev       generate reverse PAC
         bwt2sa        generate SA from BWT and Occ
         pac2cspac     convert PAC to color-space PAC
         stdsw         standard SW/NW alignment

[cem34@bulldogi bwa-0.5.7]$
```

Getting your Genome Ready

First, you will need to get your reference genome. Most genomes are here:
http://hgdownload.cse.ucsc.edu/downloads.html

Download the human genome:

```
] http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz
```

Gunzip the tarball:

```
]gunzip chromFa.tar.gz
]tar -xvf chromFa.tar
```

Remove the haplotype, unmapped, and random chromosomes

```
]rm *random*
]rm *Un*
]rm *hap*
```

Concatenate the different chromosomes

```
]cat chr*.fa >hg19.fa
```

Then, you need to build the Burrows-Wheeler Transformed Index.

```
] ./bwa index -a bwtsw hg19.fa
```

This should take 1-2 hours for human.
Now you are ready to start aligning!

```
[cem34@bulldogi bwa-0.5.7]$ ./bwa index -a bwtsw ../hg19_chrALL.fa
[bwa_index] Pack FASTA... 32.16 sec
[bwa_index] Reverse the packed sequence... 9.13 sec
[bwa_index] Construct BWT for the packed sequence...
```