From reads to results

Dr Torsten Seemann

MONASH University

Victorian Bioinformatics Consortium

vlsci • LSCC LIFE SCIENCES COMPUTATION CENTRE

What I will cover *

NGS
Applications

Sequences

Sequence quality

Read file formats

Using reads

Alignment file formats

Analysis tools

RNA-Seq
DGE

** May be different to what you remember tomorrow*

Reads

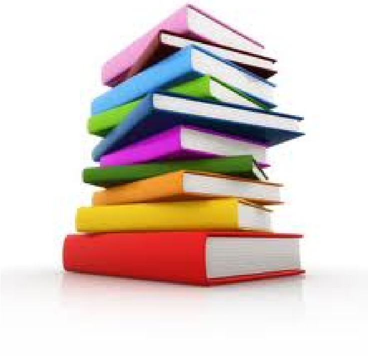
In an ideal world...

- Collect a human genomic DNA sample
- Run it through the lab sequencing machine
- Get back 46 files:
  - phased, haplotype chromosomes
  - each a single contiguous sequence of A,G,T,C
- And it only costs $1000

The awful truth

- No such instrument exists
  - can't read long stretches of DNA (yet)
- But we can read <u>short</u> pieces of DNA
  - shred DNA into ~500 bp fragments
  - we can read these reliably
- High-throughput sequencing
  - sequence millions of different fragments <u>in parallel</u>
  - various technologies to do this
  - costs much more than $1000

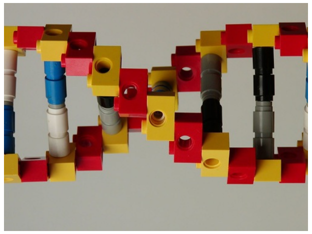
What you get back



Millions to billions of reads (big files):

```
ATGCTTCTCCGCCTTTAATTAAAATTCCATTTCGTGCACCAACACCCGTTCCTACCATAATAGCTGTTGGAGTCGCTAAACCTAATGCACATGGACACGC         <- 1st read
CTAAGATACTGCCATCTTCTTCCAACGTAAATTGTACGTGATTTTCGATCCATTTTCTTCGAGGTTCTACTTTGTCACCCATTAGTGTGGTTACTCGACG         <- 2nd read
GAATATGCGTGGACAGATGACGAATTGGCAGCAATGATTAAAAAAGTCGGCAAAGGATATATGCTACAGCGATATAAAGGACTTGGAGAGATGAATGCGG
ATCAATGCAAATACAAGATGTGACAATGCGCGCAATGCAATGATAACTGGTGTTGTCAAAAAGAAACCGAATGTCGTACCTAGTGCAACAGCCACTGCAA
GGAAAAAATGAGAAAAAATTCAGTTCGAAAACTAACGATTTCTGCTTTATTGATTGGGATGGGGGTCATTATCCCAATGGTTATGCCTAAAATCATGATC
GATGAAACAATCCAACAAATACCATTCAATAATTTCACAGGGGAAAATGAGACNCTAAGTTTCCCCGTATCAGAAGCAACAGAAAGAATGGTGTTTCGCT
...
...                                        <--- 100 bp --->
...
AGGCATCTTGAAAAAACAAGTGTGTGCCTCTGCGATAATCAATGCCACAGAGGTGCATAAAATTAGTTGTCGAAAAAATAATCGCTACCGTTGAGACTTC
AAAGGAGCATTCTTCGCACGCGGCAAAAAAAGAATACAAACGCATGTCTATAAAAGAGACAACCCAAATTACCAGACAGTTAAACGCGATTTATAAGGCT
GTGACAAAAATCGTGTCACAGCTTCTTTTATATCCTGTCTTTTTTTAGTTATTTATTTTTCAACCTTATCAATATGACTTGATAGCCTTTTCTTTTTCGA
AACTTGTTAAAAAAGACGTCAATGCCTTAACTGTACGTGATTCTTCTGCAGTTAGGGGATGACCTTTGACTACTAAAACAGATGCCATATGCTTACCTTC
ACAAAGCATATTTGTAGGAACGATTGAAAGCATCACTCAAGTAGAAGCGGAAGAAGAAACGATTCAACTGAAACTCGTCGATGTCATGGCCAAAGAAGAT
AATTGGACTTTGTCACCGATTTTCAGTTCATCTATGTCCACGCTTATTTTTTCAGCAGTAGCATTCAAAATCACTCCGTCATTGCTGAATGATGTCCCCA
CTCCTGTTTCTTTATCTATAATTGAACTGTAAACATGAGGAATCACTTTTTTTACACCTGCATCGATTGCAATTTTCAGAATTTCTTCAAAGTTTGAAAG
AAACTGCCATTCAAATGCTGCAAGACATGGGAGGTACTTCAATCAAGTATTTCCCGATGAAAGGCTTAGCACATAGGGAAGAATTTAAAGCAGTTGCGGA
ATCATTCCTACGCCAGTCATTTCGCGTAGTTCTTTTACCATTTTAGCTGTAACGTCTGCCATGTTTAACTCCTCCTGTGTGTGTTCTTTTTAAAAAAAGC        <- last read
```

Applications



If you can transform your assay in to sequencing lots of short pieces of DNA, then NGS is applicable.

Not just whole genome DNA:

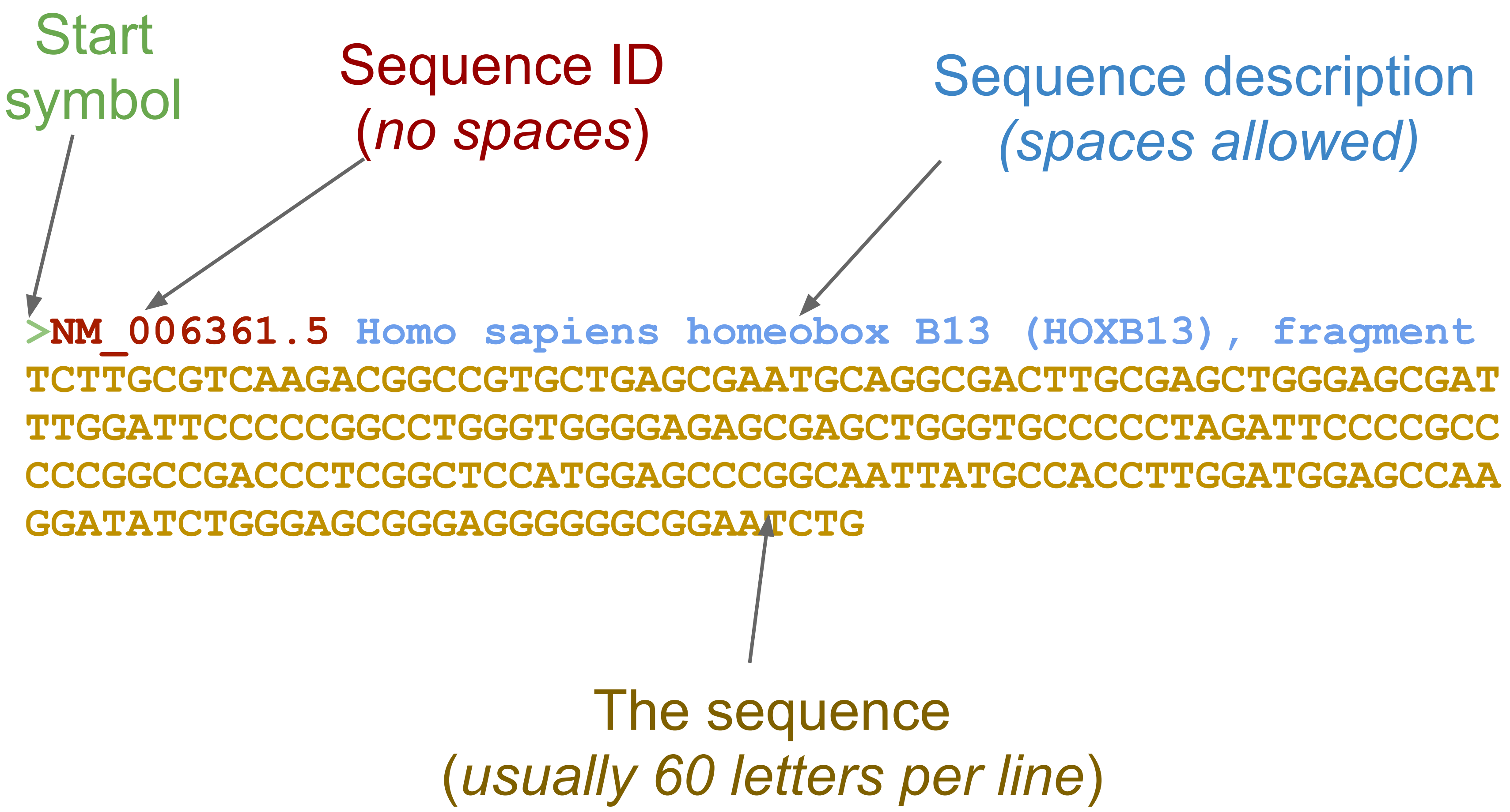
- exome (targeted subsets of genomic DNA)
- RNA-Seq (transcripts via cDNA)
- ChIP-Seq (protein:DNA binding sites)
- HITS-CLIP (protein:RNA binding sites)
- methylation (bisulphite treatment of CpG)
- ... even methods to sequence peptides now!

FASTA format

FASTA

```
>NM_006361.5 Homo sapiens homeobox B13 (HOXB13), fragment
TCTTGCGTCAAGACGGCCGTGCTGAGCGAATGCAGGCGACTTGCGAGCTGGGAGCGAT
TTGGATTCCCCCGGCCTGGGTGGGGAGAGCGAGCTGGGTGCCCCCTAGATTCCCCGCC
CCCGGCCGACCCTCGGCTCCATGGAGCCCGGCAATTATGCCACCTTGGATGGAGCCAA
GGATATCTGGGAGCGGGAGGGGGGCGGAATCTG
```

FASTA components



Start symbol

Sequence ID (*no spaces*)

Sequence description *(spaces allowed)*

```
>NM_006361.5 Homo sapiens homeobox B13 (HOXB13), fragment
TCTTGCGTCAAGACGGCCGTGCTGAGCGAATGCAGGCGACTTGCGAGCTGGGAGCGAT
TTGGATTCCCCGGCCTGGGTGGGAGAGCGAGCTGGGTGCCCCCTAGATTCCCCGCC
CCCGGCCGACCCTCGGCTCCATGGAGCCCGGCAATTATGCCACCTTGGATGGAGCCAA
GGATATCTGGGAGCGGGAGGGGGCGGAATCTG
```

The sequence (*usually 60 letters per line*)

Multi-FASTA

Concatenation of individual FASTA entries, using ">" as an entry separator

```
>read00001
TCTTGCGTCAAGACGGCCGTGCTGAGCGAATGCAGGCGACTTGCGAGCTGGGAGCGA
>read00002
TGGATTCCCCCGGCCTGGGTGGGGAGAGCGAGCTGGGTGCCCCCTAGATTCCCCGCC
>read00003
GGCCGACCCTCGGCTCCATGGAGCCCGGCAATTATGCCACCTTGGATGGAGCCAAGG
>read00004
TCTGGGAGCGGGAGGGGGCGGAATCTGGAGCGAGCTGGGTGCCCCCTAGATTCCCC
>read00004
GCGGAATCTGGAGCGAGCTGGGTGCCCCCTAGATTCCCCGCATCGTAGATTAGATAT
```

The DNA alphabet

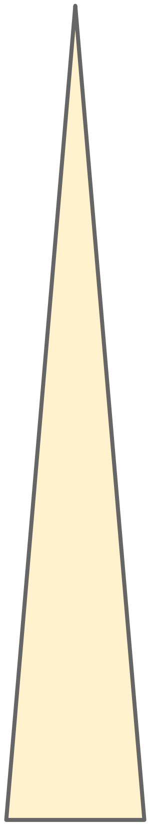
- Standard
  - A G T C
- Extended
  - adds N (unknown base)
- Full
  - adds R Y M S W K V H D B (ambiguous bases)
  - R = A *or* G (puRine)
  - Y = C *or* T (pYrimidine)
  - *... and so on for all the combinations*

Sequence Quality

Sequences have errors

- nonsense reads
  - *instrument oddness*
- duplicate reads
  - *amplify a low complexity library*
- adaptor read-through
  - *fragment too short*
- indel errors
  - *skipping bases, inserting extra bases*
- uncalled base
  - *couldn't reliably estimate, replace with "N"*
- substitution errors
  - *reading wrong base*

Less common



More common

Illumina reads

- Usually 100 bp (soon 250 bp)
- Indel errors are rare
- Substitution errors < 1%
  - Error rate higher at 3' end
- Adaptor issues
  - rare in HiSeq (*TruSeq* prep)
  - more common in MiSeq (*Nextera* prep)
- Very high quality overall

DNA base quality

REJECTED

- DNA sequences often have a *quality value* associated with each nucleotide
- A measure of reliability for each base
  - as it is derived from physical process
    - chromatogram (Sanger sequencing)
    - pH reading (Ion Torrent sequencing)
- Formalised by the *Phred* software for the Human Genome Project

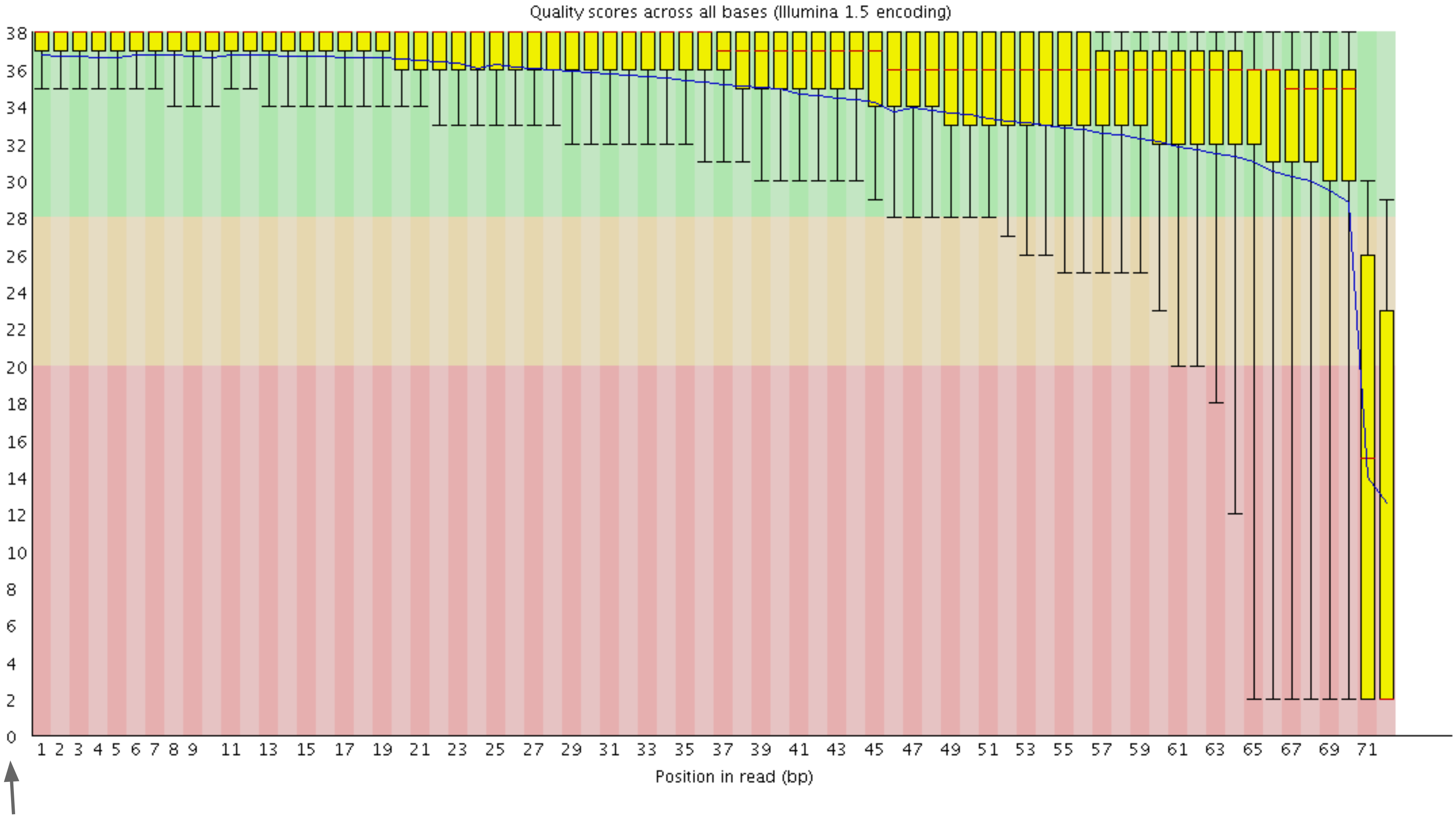
Phred qualities

Quality	Chance it's wrong	Accuracy	Description
10	1 in 10	90%	Bad
20	1 in 100	99%	Maybe
30	1 in 1000	99.9%	OK
40	1 in 10,000	99.99%	Very good
50	1 in 100,000	99.999%	Excellent

$$Q = -10 \log_{10} P \quad \Leftrightarrow \quad P = 10^{-Q/10}$$

Q = Phred quality score P = probability of base call being incorrect

Quality plot (*FastQC*)



Y-axis is "Phred" quality values (higher is better)

Quality filtering

- Keep all reads
  - let the downstream software cope
- Reject some reads
  - average quality below some threshold
  - contain any ambiguous bases
- Trim reads
  - remove low quality bases from end
  - keep longest "sub-read" that is acceptable
- Best strategy is analysis dependent

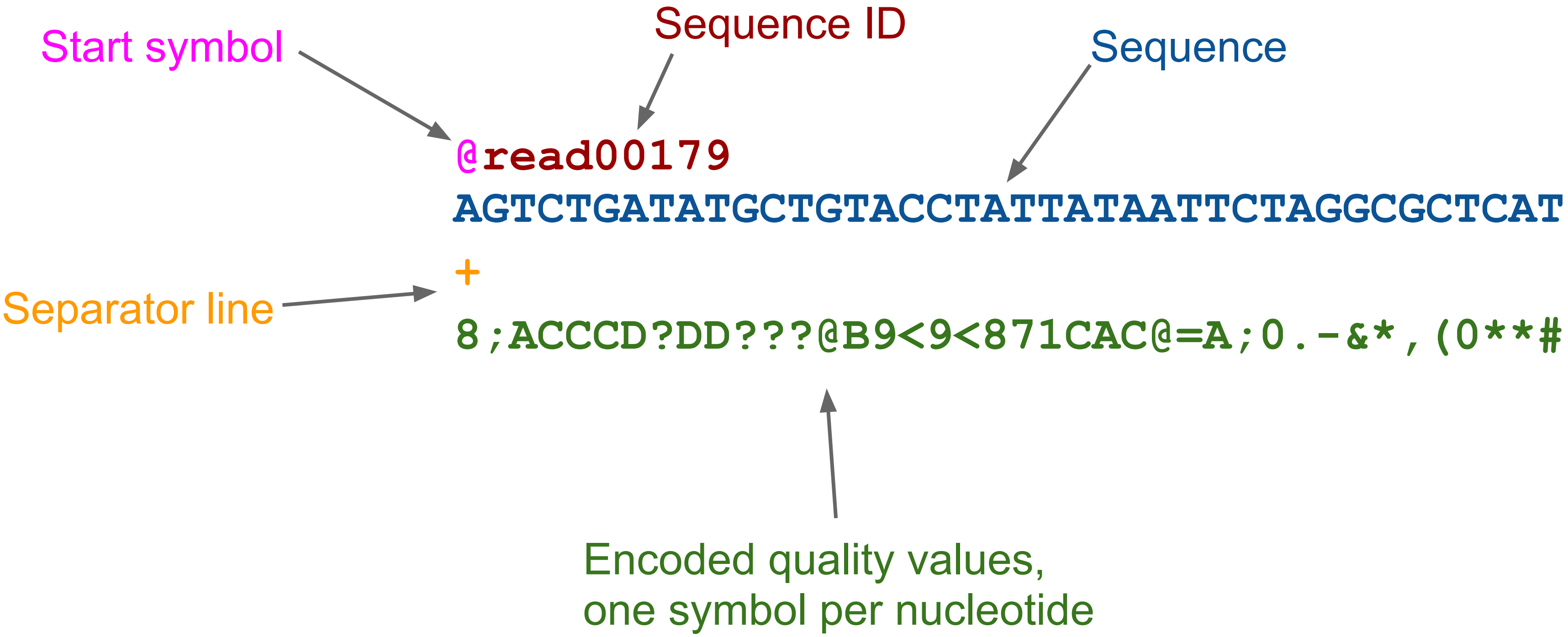
FASTQ files

FASTQ

A sequence read looks like this:

```
@read00179
AGTCTGATATGCTGTACCTATTATAATTCTAGGCGCTCAT
+
8;ACCCD?DD???@B9<9<871CAC@=A;0.-&*,(0**#
```

FASTQ components



Start symbol

Sequence ID

Sequence

```
@read00179
AGTCTGATATGCTGTACCTATTATAATTCTAGGCGCTCAT
+
8;ACCCD?DD???@B9<9<871CAC@=A;0.-&*,(0**#
```

Separator line

Encoded quality values,
one symbol per nucleotide

FASTQ quality encoding

Uses letters/symbols to represent numbers:

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
|         |         |         |         |
Q0        Q10       Q20       Q30       Q40
bad     maybe       ok      good    excellent
```

Multi-FASTQ

Same as multi-FASTA, just concatenate:

```
@M00267:3:000000000-A0AGE:1:1:15997:1501
CTCGTGCTCTACTTTAGAAGCTAATGATTCTGTTTGTAGAACATTTTCTACCACTACATCTTTTTCTTGCTTCGCATCTT
+
:=?DD:BDDF>FFHI>E>B9AE>4C<4CCAE+AEG3?EAGEHCGIIIIIIIIIIIIGIIIEIIIIGGIDGIID/;4C<EE
@M00267:3:000000000-A0AGE:1:1:15997:1501
GCCTATAGTAGAAGAAAAAGAAGTGGCTCAAGAAATGAGTGCACCGCAGGAAGTTCCAGCGGCTGAATTACTTCATGAAA
+
<@@FFF?DHFHGHIIIFGIIGIGICDGEGCHIIIIIIIIIGIHIIFG<DA7=BHHGGIEHDBEBA@CECDD@CC>CCCAC
@M00267:3:000000000-A0AGE:1:1:14073:1508
GTCTTGCTAAATTTAAATAATCTGAAATAATTTGTTCTGCCCGGTCCAATTCAGCTAATACGAGACGCATATAATCCTTA
+
:?DDDDD?84CFHC><F>9EEH>B>+A4+CEH4FFEHFHIIIIIIIIIIIIIGGIIIIIIIIG>B7BBEBBB@CDDCFC
@M00267:3:000000000-A0AGE:1:1:14073:1508
ACGTACAGAGATGCAAAAGTCAGAGAAACTTAATATTGTAAGTGAGTTAGCAGCAAGTGTTGCACATGAGGTTCGAAATC
+
1@@DDDADHGDF?FBGGAFHHCHGGCGGFHIECHGIIGIGFGHGHIIHHEGCCFCB>GEDF=FCFBGGGD@HEHE9=;AD
```

Data compression

- FASTQ files are very big
  - typically > 10 gigabytes
  - they are somewhat redundant
- Often they will be compressed
  - gzip (.gz extension)
  - bzip2 (.bz2 extension)
  - these are like .ZIP but different method
- Usually get to < 20% of original size
  - faster transfer, less disk space
  - can be slower to read and write though

FASTQ file name conventions

Suffix	Usage
.fastq .fq	Uncompressed
.fastq.gz .fq.gz	Compressed with GZIP
.fastq.bz2 .fq.bz2	Compressed with BZIP2
s_?_?_sequence.txt	Old Illumina naming (uncompressed)

Using reads

In the beginning

First step of most NGS analyses is either:

- ***De novo* assembly**
  - reconstruct the original sequences from reads alone
  - like a jigsaw puzzle but ambiguous
- **Align to reference**
  - find where reads fit on a known sequence
  - can not always be uniquely placed

De novo assembly

De novo assembly

- *Reconstruct the original DNA sequences using the sequence reads alone*

- Method
  - align each read against every other read
  - build an overlap graph
  - simplify the graph (necessary due to read errors)
  - find distinct paths through the graph
  - calculate the consensus sequence for each path
  - these contiguous sequences are <u>contigs</u>

- Computationally challenging problem

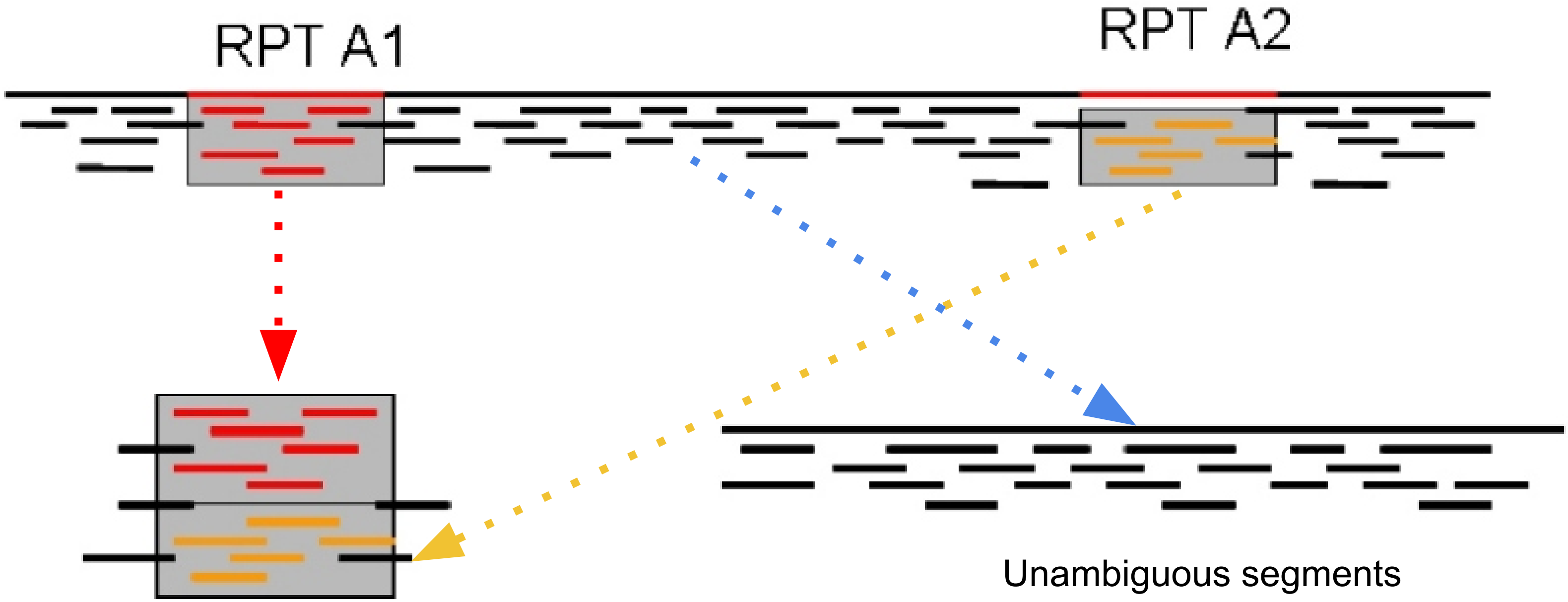
Why *de novo* ?

- Sequence a new organism
  - DNA-Seq (genome, expect novel DNA)
  - RNA-Seq (transcriptome, splice variants)
- Unaligned reads from reference alignment
  - novel DNA segments
  - novel RNA transcripts
  - fusion genes
  - contamination

Where assembly "fails"

WANTED
IN 5 STATES
JOHN DILLINGER
PUBLIC ENEMY
NUMBER ONE!
The Attorney General of the U.S. has authorized a
$20,000 REWARD!
for information leading to the arrest of John Dillinger.

It is impossible to resolve (disambiguate) repeats of length L with reads shorter than L



Repeats get collapsed
into a single contig

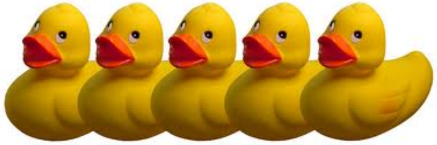
Unambiguous segments
get their own contigs

Assembly file format

- Usually a simple multi-FASTA file of contigs
  - some assemblers provide quality values
  - sequences may contain gaps of "N"s
- Loss of information
  - contigs are a "collapsing" of the rich graph structure
  - ambiguous, but useful connective info is lost
- New standard coming
  - retain all graph information
  - will allow development of new post-processing tools

Align to reference

NGS read alignment



- Want to find where all the reads fit on our reference genome, quickly and accurately.
- Query
  - Lots (>100M) of short (~100bp) reads (FASTQ)
- Reference
  - *eg.* Human genome, ~27000 contigs (FASTA)
- Many shorts *vs.* few longs
  - BLAST isn't suitable (it's better at the opposite)
  - New tools: BWA, Bowtie, BFAST, SHRiMP, MAQ

Example

Seven short 4bp reads:

AGTC TTAC GGGA CTTT TAGG TTTA ATAG

The 31bp reference:

```
AGTCTTTATTATAGGGAGCCATAGCTTTACA
AGTC      TAGG     ATAG   TTAC
    TTTA     GGGA       CTTT
```

Coverage:

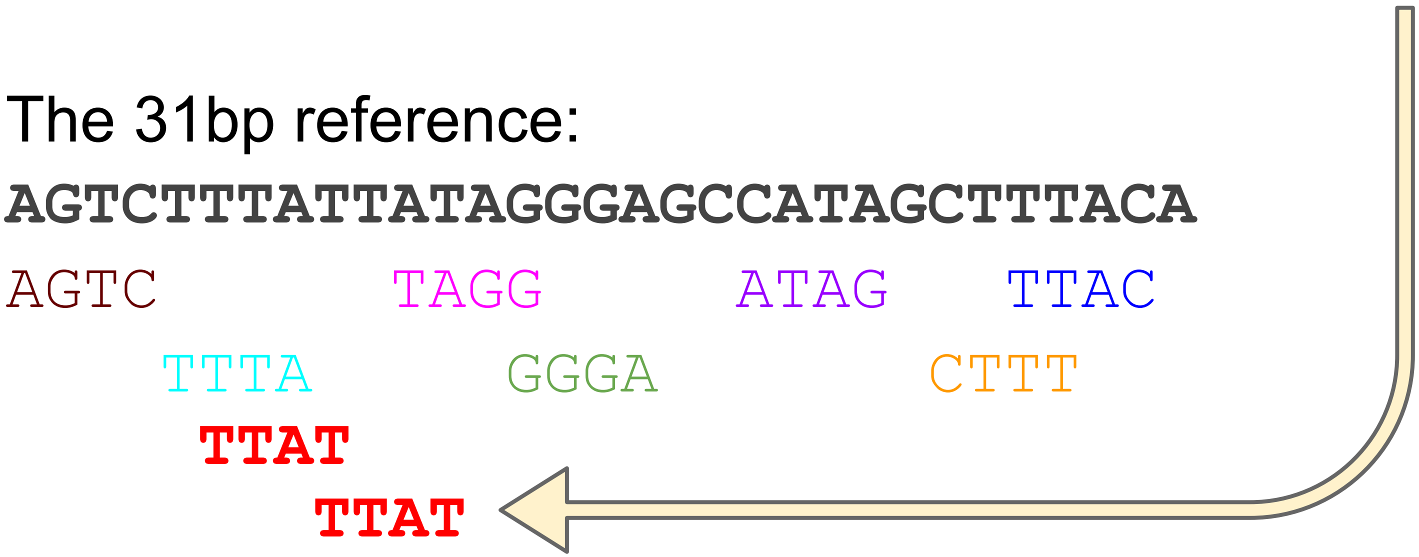
1111111100111211100111101122110

Average coverage (depth): 28/31 = 0.90x

Ambiguous alignment

Eight short 4bp reads:

AGTC TTAC GGGA CTTT TAGG TTTA ATAG **TTAT**



The 31bp reference:

```
AGTCTTTATTATAGGGAGCCATAGCTTTACA
AGTC      TAGG     ATAG   TTAC
    TTTA     GGGA       CTTT
     TTAT
        TTAT
```

Reads can align to more than one place!

Multiple-mapping reads

- Align to all possible places
  - useful in some situations
  - but belongs to only one "real" place

- Align to the first place you find
  - not a good idea... but some tools still do it

- Align to a random choice of all valid places
  - useful in some situations

- Don't use multiple-mapping reads
  - often necessary if calling SNPs

The trade-off

- speed vs. sensitivity
- will miss divergent matches
- can miss indels (insertions and deletions)

BAM files

Storing alignments

- SAM
  - plain text file, tab separated columns
  - "a huge spreadsheet"
  - inefficient to read and store
- BAM
  - a compressed version of SAM (~80% less storage)
  - can be indexed (fast access to subsections)
  - needs to be sorted to be useful however
- Standardized format
  - readable by most software

What's in a SAM/BAM?

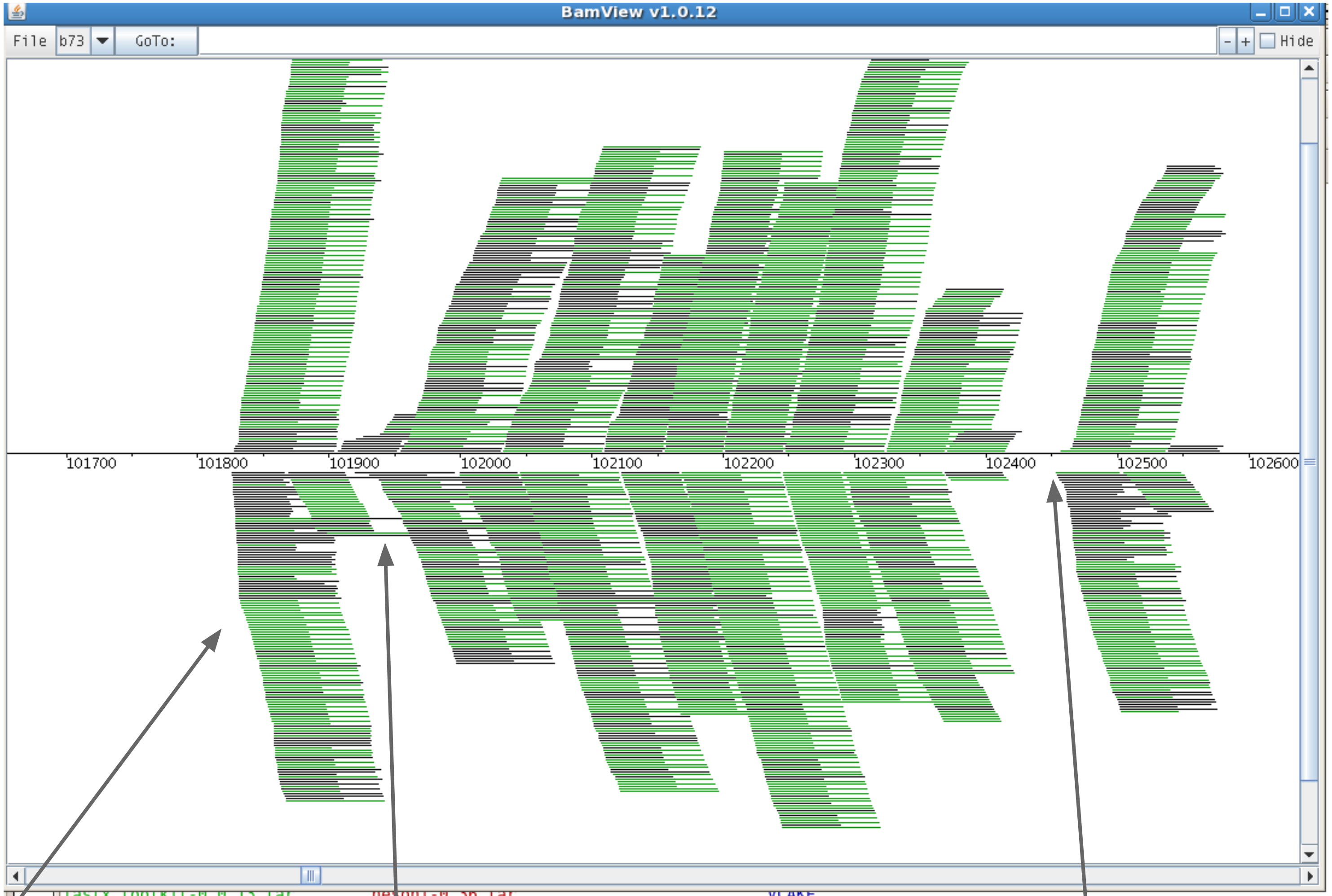
```
1:497:R:-272+13M17D24M   113  1    497  37   37M  15   100338662 0
CGGGTCTGACCTGAGGAGAACTGTGCTCCGCCTTCAG   0;==-==9;>>>>>=>>>>>>>>>>>>=>>>>>>>>>
XT:A:U    NM:i:0    SM:i:37   AM:i:0    X0:i:1    X1:i:0    XM:i:0

19:20389:F:275+18M2D19M  99   1    17644     0    37M  =    17919     314
TATGACTGCTAATAATACCTACACATGTTAGAACCAT   >>>>>>>>>>>>>>>>>>>>><>>><<>>4::>>:<9   RG:Z:
UM0098:1  XT:A:R    NM:i:0    SM:i:0    AM:i:0    X0:i:4    X1:i:0    XM:i:0

19:20389:F:275+18M2D19M  147  1    17919     0    18M2D19M  =    17644     -314
GTAGTACCAACTGTAAGTCCTTATCTTCATACTTTGT   ;44999;499<8<8<<<8<<><<<<><7<;<<<>><<   XT:A:R
     NM:i:2    SM:i:0    AM:i:0    X0:i:4    X1:i:0    XM:i:0    MD:Z:18^CA19
```

- One line per original read sequence
  - where it aligned (if at all)
  - how much of it aligned (soft/hard clipping)
  - how well it aligned (mapping quality)
  - any differences to the reference (CIGAR string)
  - lots of other stuff (aligner dependent)

Wide view (BamView)

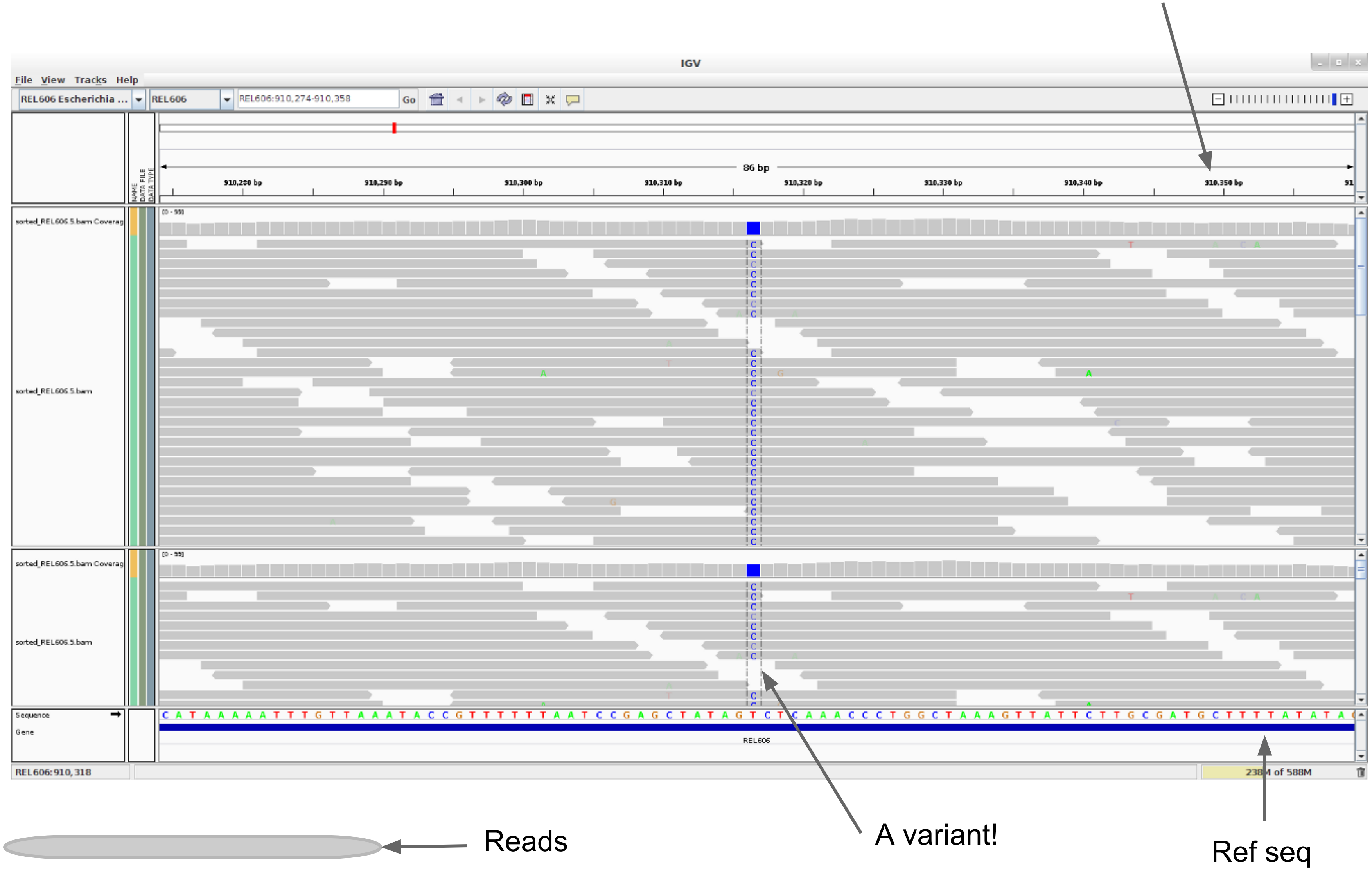


High coverage

Low coverage

Zero coverage

Medium view (IGV)



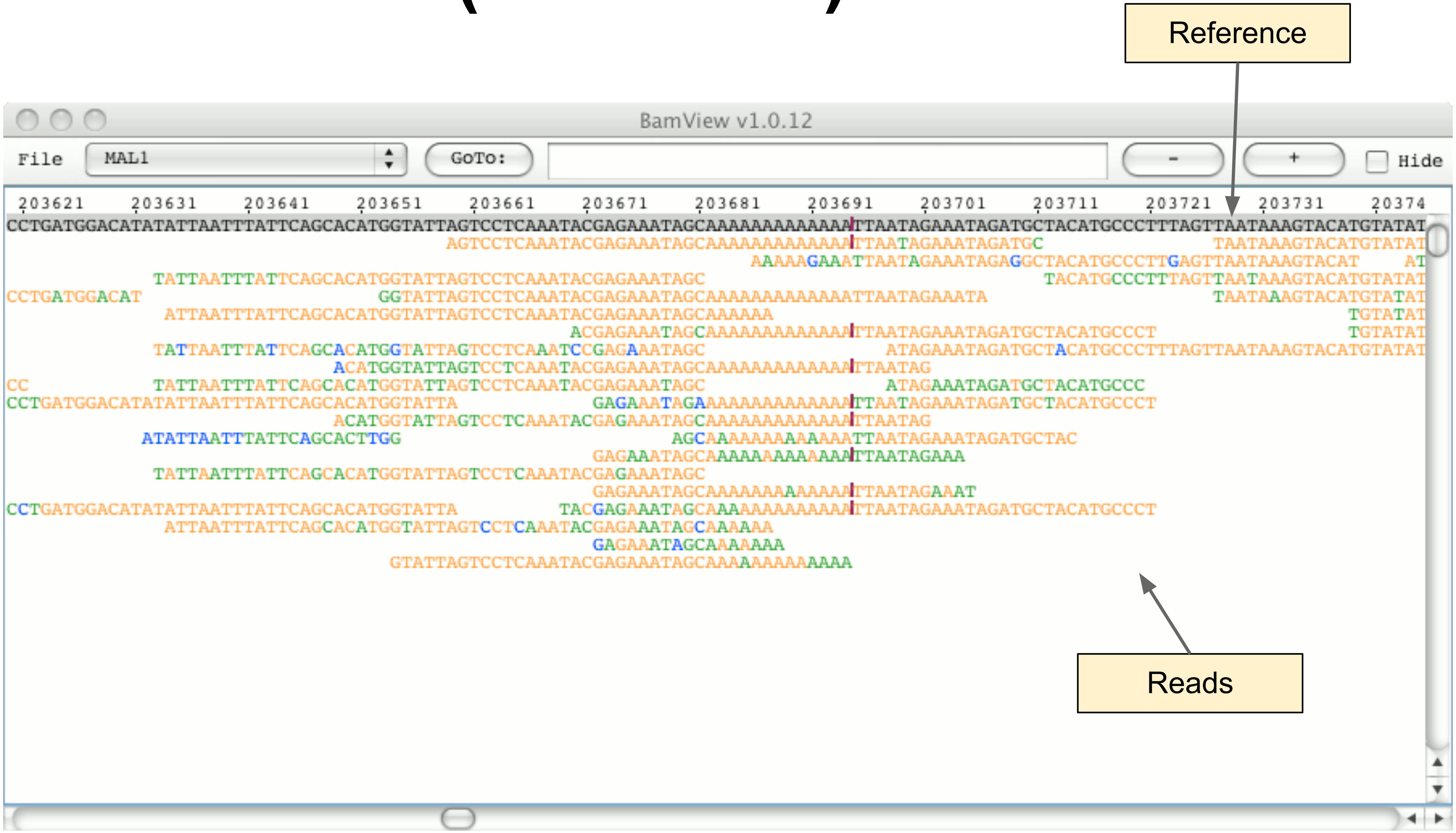
Reference coordinates

Reads

A variant!

Ref seq

Close view (BamView)



```
 203621    203631    203641    203651    203661    203671    203681    203691    203701    203711    203721    203731    20374
CCTGATGGACATATATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAATTAATAGAAATAGATGCTACATGCCCTTTAGTTAATAAAGTACATGTATAT
                                       AGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAATTAATAGAAATAGATGC               TAATAAAGTACATGTATAT
                                                                  AAAAAGAAATTAATAGAAATAGAGGCTACATGCCCTTGAGTTAATAAAGTACAT    AT
             TATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGC                              TACATGCCCTTTAGTTAATAAAGTACATGTATAT
CCTGATGGACAT                     GGTATTAGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAATTAATAGAAATA                    TAATAAAGTACATGTATAT
              ATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGCAAAAAA                                                   TGTATAT
                                                  ACGAGAAATAGCAAAAAAAAAAAAATTAATAGAAATAGATGCTACATGCCCT                 TGTATAT
             TATTAATTTATTCAGCACATGGTATTAGTCCTCAAATCCGAGAAATAGC                ATAGAAATAGATGCTACATGCCCTTTAGTTAATAAAGTACATGTATAT
                             ACATGGTATTAGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAATTAATAG
CC           TATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGC                ATAGAAATAGATGCTACATGCCC
CCTGATGGACATATATTAATTTATTCAGCACATGGTATTA            GAGAAATAGAAAAAAAAAAAAAATTAATAGAAATAGATGCTACATGCCCT
                             ACATGGTATTAGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAATTAATAG
            ATATTAATTTATTCAGCACTTGG                        AGCAAAAAAAAAAAAATTAATAGAAATAGATGCTAC
                                                    GAGAAATAGCAAAAAAAAAAAAATTAATAGAAA
             TATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGC
                                                    GAGAAATAGCAAAAAAAAAAAAATTAATAGAAAT
CCTGATGGACATATATTAATTTATTCAGCACATGGTATTA         TACGAGAAATAGCAAAAAAAAAAAAATTAATAGAAATAGATGCTACATGCCCT
              ATTAATTTATTCAGCACATGGTATTAGTCCTCAAATACGAGAAATAGCAAAAAA
                                                    GAGAAATAGCAAAAAAA
                                  GTATTAGTCCTCAAATACGAGAAATAGCAAAAAAAAAAAAA
```

Getting results

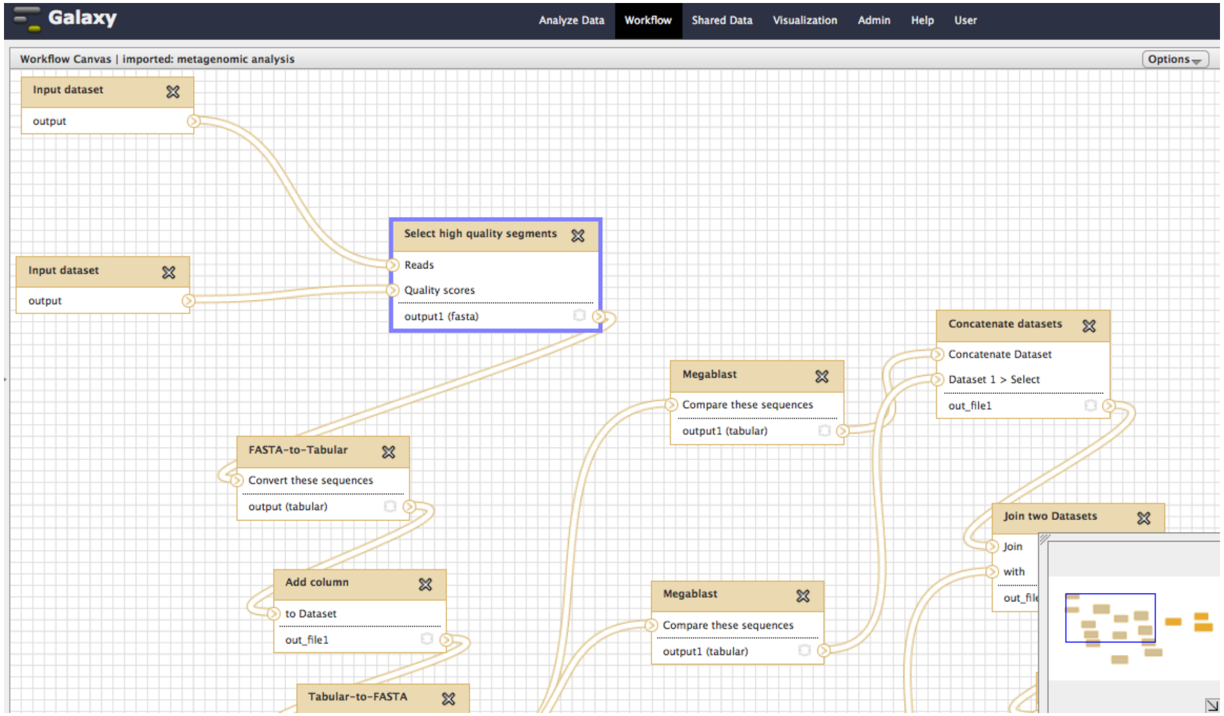
Analysis

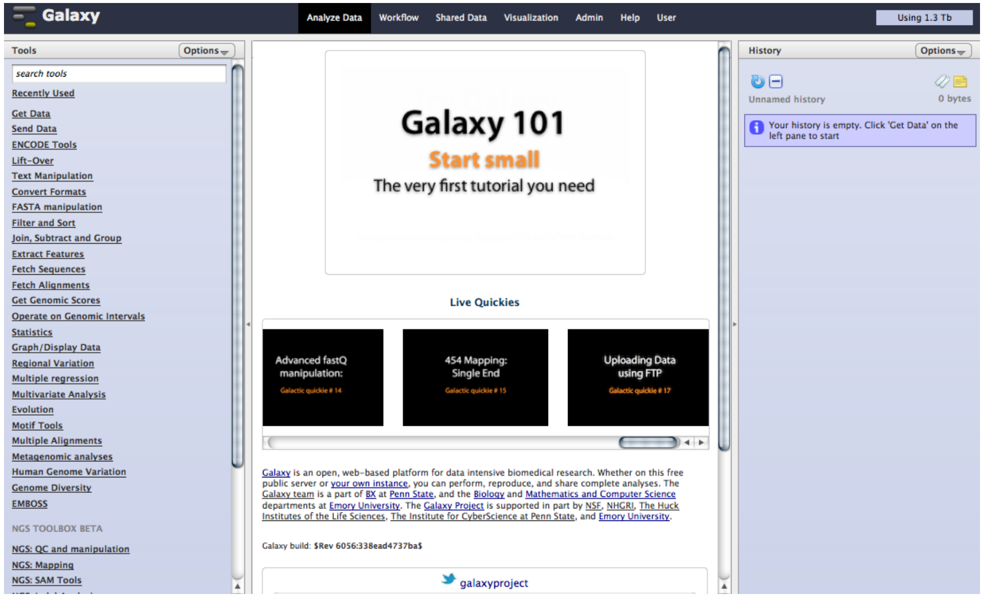
- Desktop
  - Free - need Unix skills or a bioinformatician
  - Commercial - CLC Genomics Workbench

- Server / Cloud
  - Free - Galaxy / Genomics Virtual Laboratory
  - Commercial - Illumina Basespace, ...

- Service / Subscription
  - AGRF, QFAB, Geneworks, ...
  - VLSCI::LSCC - hire team at 0.5 EFT etc

Galaxy

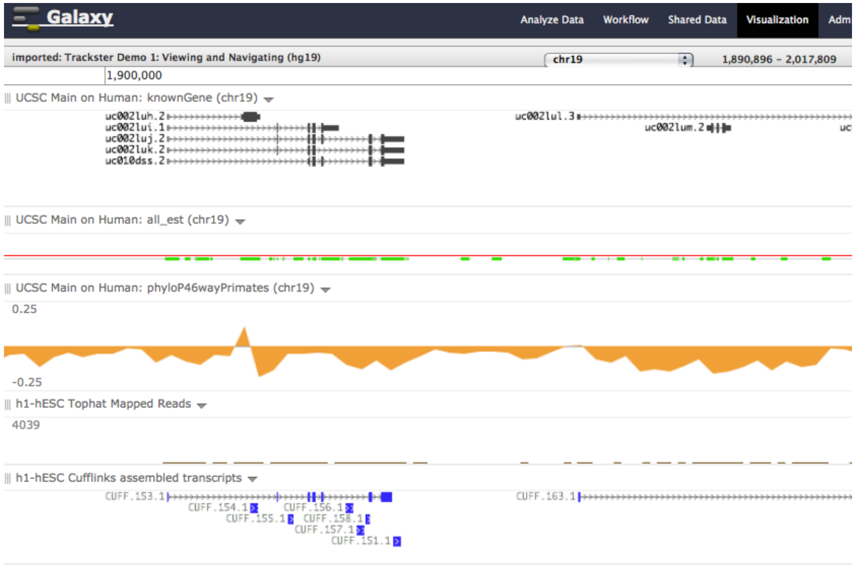




Your history is empty. Click 'Get Data' on the left pane to start

Galaxy is an open, web-based platform for data intensive biomedical research. Whether on this free public server or your own instance, you can perform, reproduce, and share complete analyses. The Galaxy team is a part of BX at Penn State, and the Biology and Mathematics and Computer Science departments at Emory University. The Galaxy Project is supported in part by NSF, NHGRI, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Emory University.

Galaxy build: $Rev 6056:338ead4737ba$



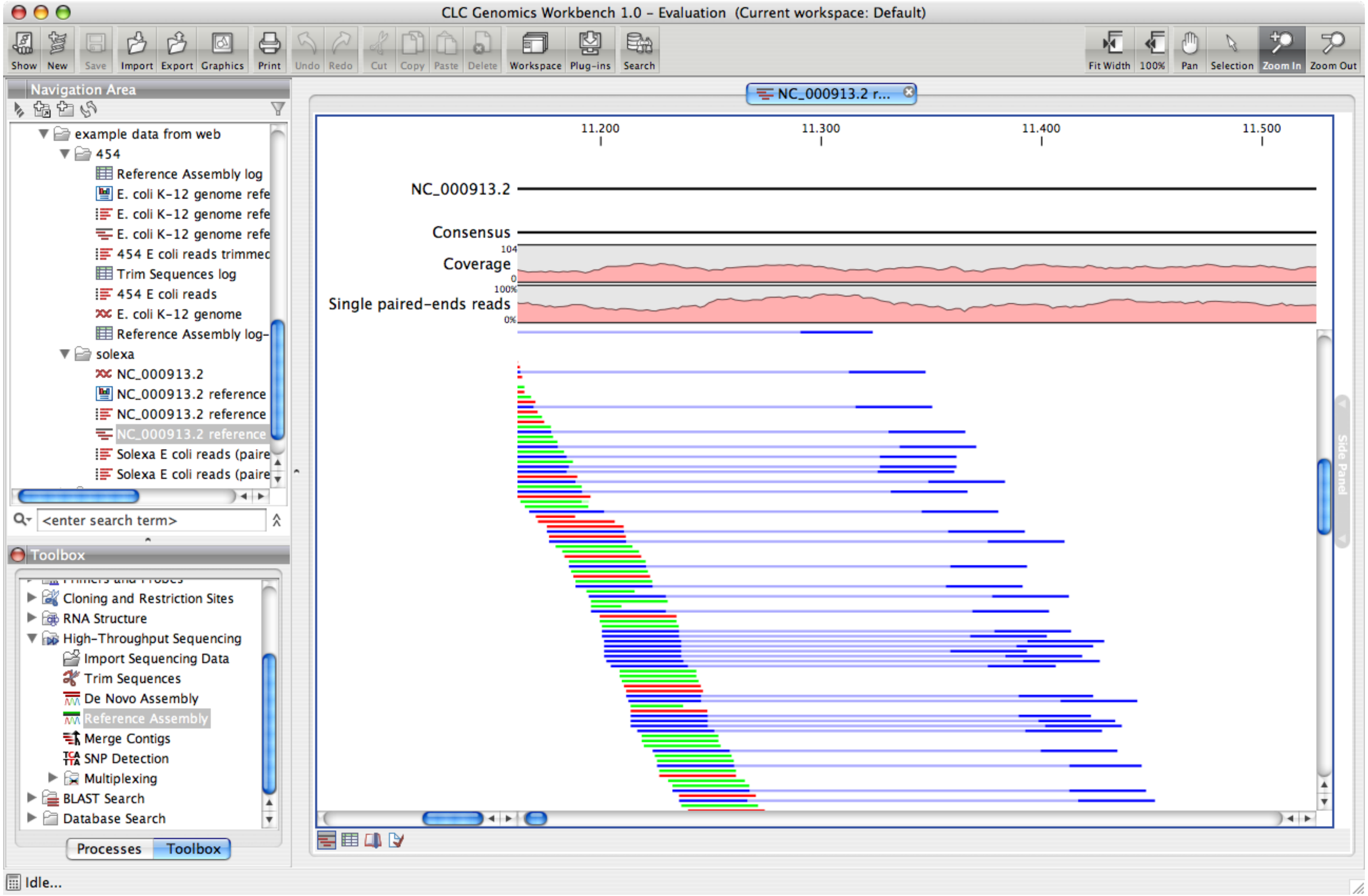
Galaxy

- Web-based
  - install on your own server
  - install via Cloudman onto your Amazon EC2 cloud
  - use the free public server at *usegalaxy.org*

- Lots of tools and workflows available
  - getting close to plug and play

- Coming soon...
  - Australian Research Cloud
  - will host the Genomics Virtual Laboratory
  - you will be able to run Galaxy instances on it

CLC Genomics Workbench



CLC Genomics Workbench

- Designed for NGS analysis
    - Handles most common analyses
        - alignment, RNA-Seq DGE, BLAST, assembly ...
    - but results aren't as good as bioinformatician
        - using Unix and R tools

- Accessible
    - Intuitive interface
    - Runs on Windows, Mac, Linux
    - Needs powerful desktop - lots of RAM
    - "Affordable" licences ~$4000 (highly variable)

Summary

Key points

- Getting into NGS is not trivial
  - new thinking, methods, hardware, software
- Understand the main file types
  - FASTA, FASTQ, SAM, BAM
- Repeat regions cause lots of problems
  - hard/impossible to assemble
  - multi-mapping reads
- No solution fits all problems
  - May need to collaborate/employ bioinformaticians

That's all Folks!