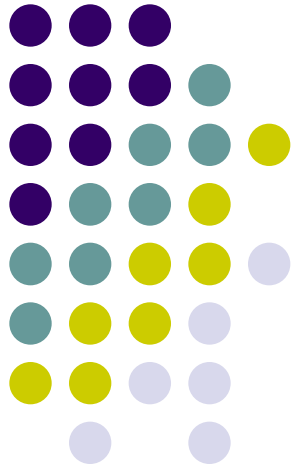
Introduction to NGS Analysis

October, 2015

Anna Battenhouse
Associate Research Scientist
abattenhouse@utexas.edu

Amelia Weber Hall
Graduate Student
ameliahall@utexas.edu

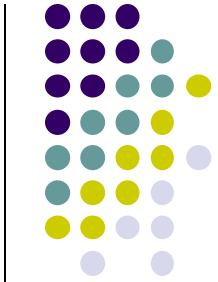
Vishwanath Iyer Lab
Center for Systems and Synthetic Biology (CSSB)
The University of Texas at Austin



Goals

- Introduce NGS vocabulary
  - provide both high-level view and important consideration details
- Focus on common, initial tasks
  - raw sequence preparation, alignment to reference
  - common bioinformatics tools & file formats
- Understand required skills & resources
  - computational & storage resources
  - highlight best practices

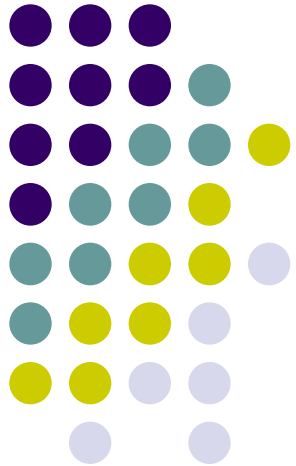
Outline



- Introductions
- Challenges & best practices
- NGS overview & terminology
- The FASTQ format
- Raw data QC and preparation
- Alignment to a reference

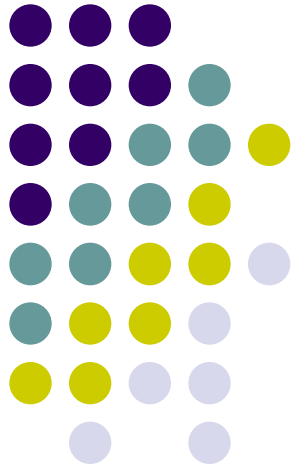
Other NGS Resources at UT

- CCBB Summer School courses
  - 4 half-day sessions
  - Intro to NGS, RNAseq, many others
  - lots of hands-on, including w/TACC
- Genome Sequencing & Analysis Facility (GSAF)
- CCBB Bioinformatics consultants



Introductions

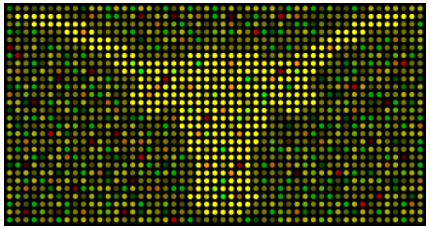
- The Iyer lab
- Our background
- Your background

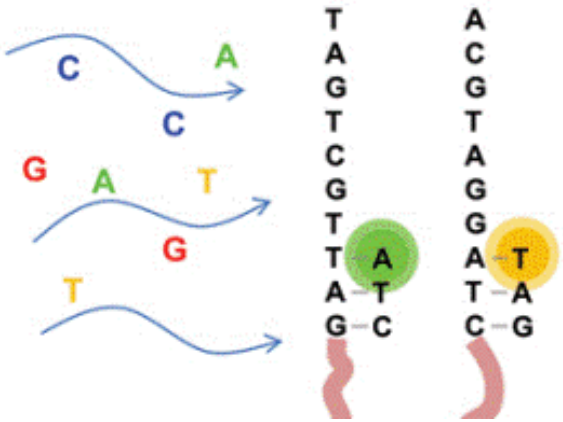


The Iyer Lab

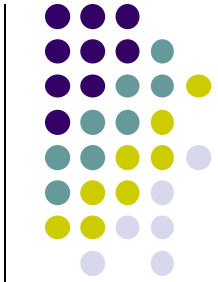
Vishy Iyer, PI

- Main focus is functional genomics
  - large-scale transciptional regulation and reprogramming in response to diverse stimuli
- Model organisms
  - human, yeast
- Research methods include
  - microarrays
  - high-throughput sequencing
    - now have > 1,700 NGS datasets
      - ChIP-seq
      - RNA-seq, RIP-seq, miRNAseq…
      - MNase-seq, MNase-ChIP-seq…
  - ENCODE consortium collaborator



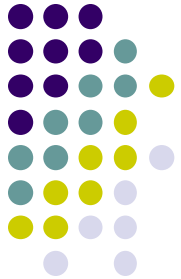


Anna's background



- Non-traditional path
  - BA English literature, 1978
  - commercial software development 1982–2005
    - Texas Instruments, Motorola…
  - joined Iyer Lab 2007 ("retirement career")
  - BS Biochemistry, 2013
- Carry-overs from industry
  - large systems & datasets
  - emphasis on process & reproducibility
    - automation
    - naming conventions
    - history checkpoints (version control, data backups)
    - component testing
    - documentation, documentation, documentation

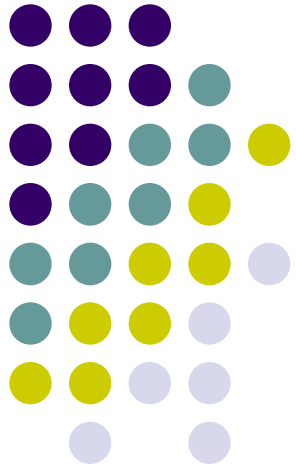
Amelia's background



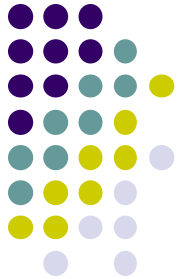
- 6th year Microbiology graduate student
  - Lab Technician in the Aldrich Lab (UT Neurobiology), 2007-2010
  - BS Molecular Genetics, 2007
- 10 years as a molecular biologist
  - Current Research: Exploring how histone modifications affect transcription in glioblastoma multiforme
  - Extensive experience preparing NGS libraries (2012-present)
    - especially ChIP-seq, also 4C
    - developed Iyer Lab protocols for dealing with solid tumor tissue
- Some programming experience (not much!)
  - Programming Intern in college (2004-2006)
  - Educational Technologies Lab IT staff (2003-2007)

NGS Analysis Challenges

- Broad skill set requirements
- Large and growing datasets
- Best practices



Diverse skill requirements

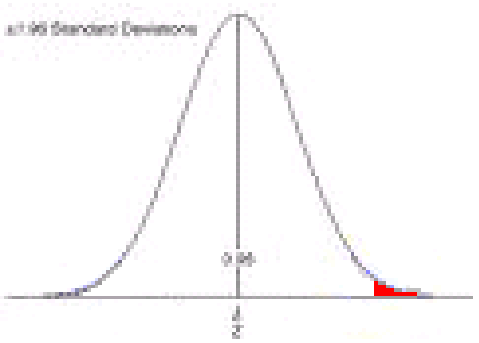


- Analysis – making sense of raw data
  - one part bioinformatics & statistics
  - one part programming / scripting
    - R, Python, Perl
    - Unix (Linux) command line (**grep**, **awk**, **sed**)
    - bash scripting
- Management – making order out of chaos
  - one part organization
  - one part data wrangling
- Adoption of best practices is critical!

Other useful traits ☺

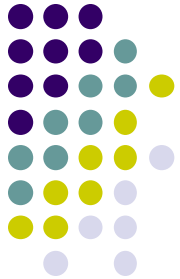
- Love of spreadsheets

- Being 3+ standard deviations above the mean on a Type A / obsessive-compulsive scale



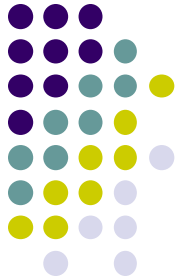
- Experiencing physical discomfort upon viewing a disorderly directory structure

NGS Analysis Best Practices



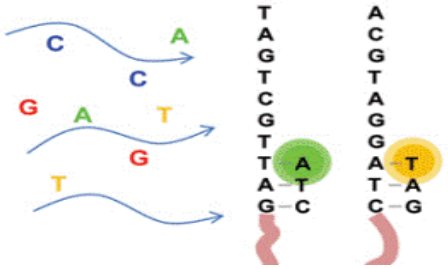
- Track your analysis steps
  - your analyses should be reproducible by others
  - need to keep the equivalent of a lab notebook to document analysis protocols
    - “work files” that detail all analysis steps performed
- Find a supercomputer
  - UT’s TACC provides computation, storage, consulting
    - no-cost for allocations for UT researchers
    - allocations for external researchers also available

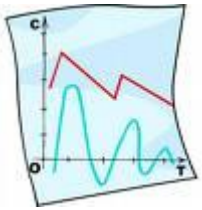
Large and growing datasets

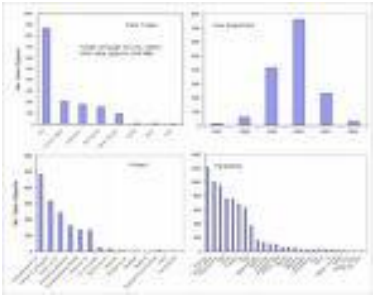


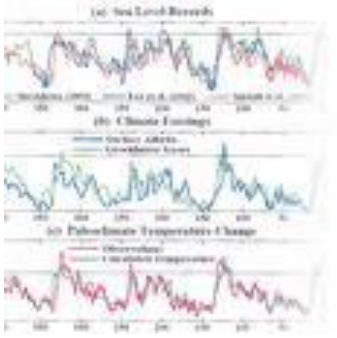
- Major Iyer Lab NGS datasets over time
  - 2008 – Shivaswamy *et. al* – Yeast heat shock remodeling
    - 2 yeast MNase-seq datasets
    - less than 2 million reads
  - 2010 – McDaniell *et. al* – CTCF binding allelic bias
    - 13 CTCF ChIP-seq datasets from 6 cell lines
    - ~ 200 million sequences
  - 2012 – Lee *et. al* – ENCODE2 data analysis
    - 32 ChIP-seq datasets: 3 transcription factors in 11 cell lines
    - ~ 1 billion sequences
  - 2013 – Polioudakis *et. al* – miRNA overexpression analysis
    - 42 RNAseq datasets: 7 conditions, Ago IP + total RNA, 3 replicates
    - ~ 2.6 billion sequences
  - 2014 – Ding, Ni, Timmer *et. al* – eQTL analysis of CTCF binding
    - 53 CTCF ChIP-seq datasets from 53 individuals
    - ~ 8.3 billion reads
  - 2015 – genome-wide regulatory changes in gliobastoma (largely complete)
    - > 400 datasets (ChIP-seq, RNAseq, miRNAseq, whole genome, exome capture, 4c)
    - ~ 20 billion sequences

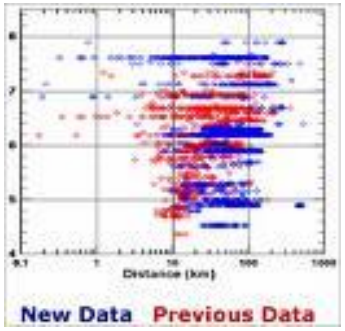
(Big) Data Wrangling

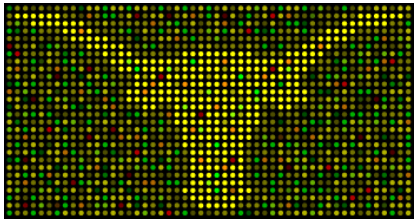




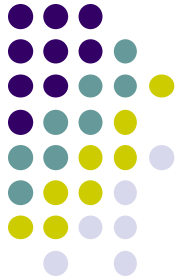






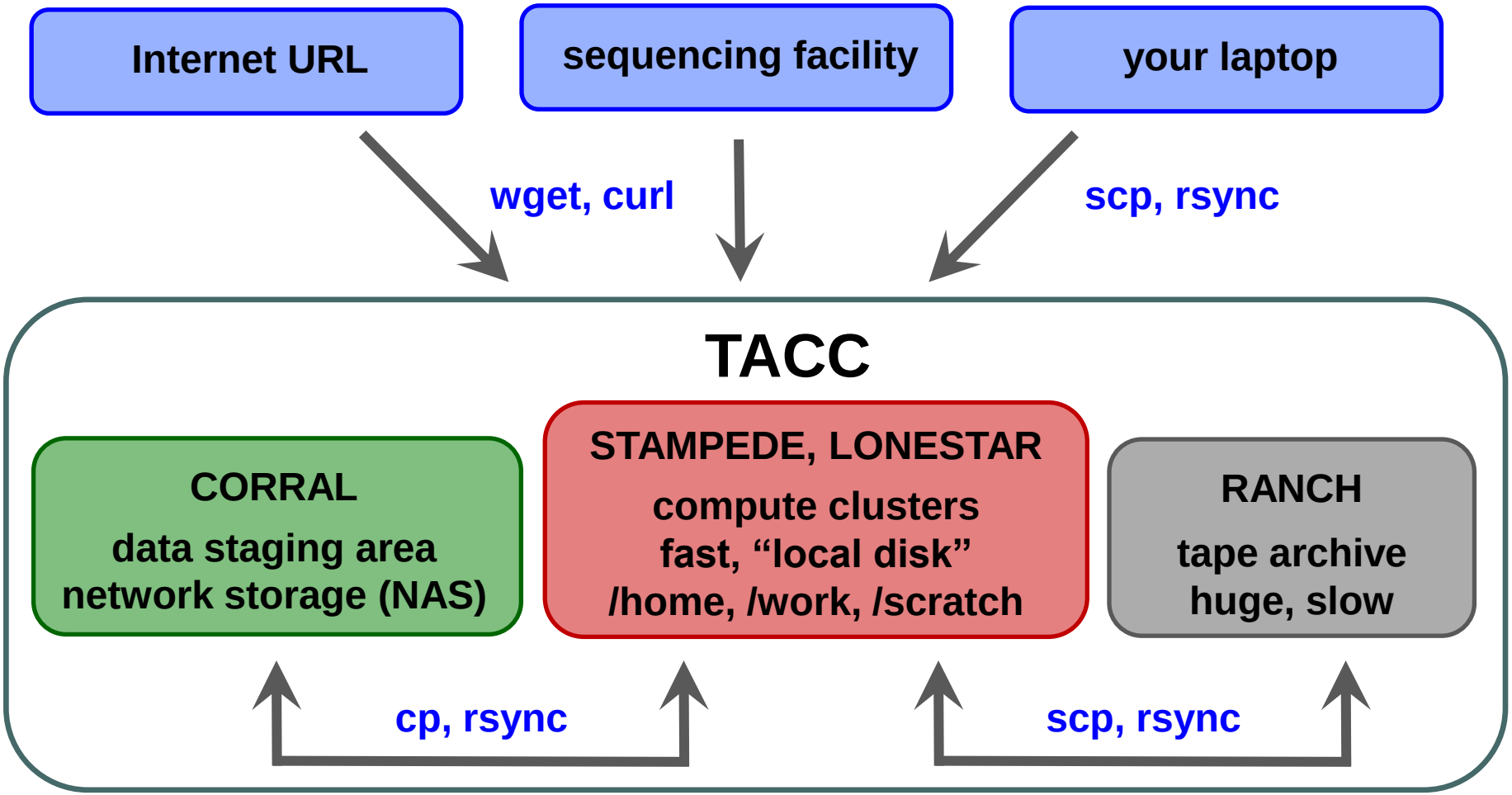


Data Wrangling Best Practices

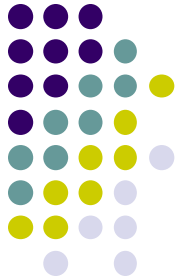


- Arrange adequate storage space
  - obtain an allocation on TACC's **corral** disk array
    - initial 5 TB are no-cost
  - stage active projects on **corral**
    - copy data to $WORK or $SCRATCH for analysis
    - copy important analysis products back to **corral**
    - periodically back up **corral** directories to **ranch** tape archive
- Back up analysis artifacts regularly
  - obtain an allocation on TACC's **ranch** tape archive system
    - 10 TB a good initial number
    - free! and under-utilized
  - periodically back up **corral** directories to **ranch** tape archive

TACC Data Flow Overview



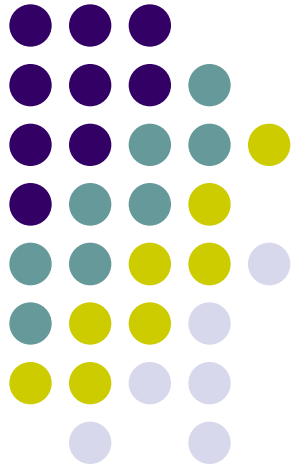
Data Wrangling Best Practices

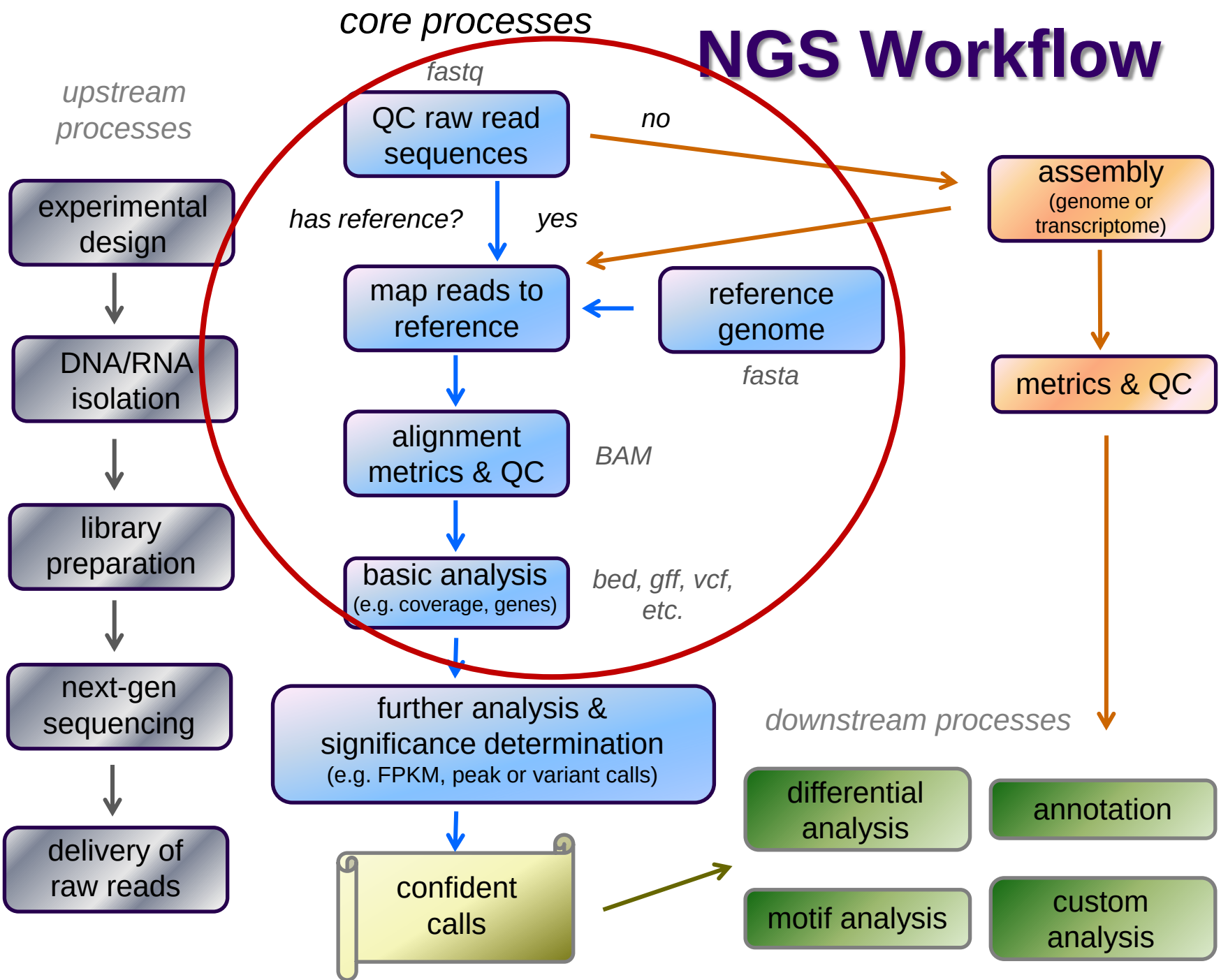


- Distinguish between types of data
  - original sequence data (FASTQ files) must be backed up!
  - alignment files are even larger than original FASTQs
    - should be backed up once stable
  - downstream analysis artifacts
- Learn to work with binary file formats
  - gzip, BAM, bigBed, bigWig
  - techniques exist for working with them without converting files back to uncompressed text

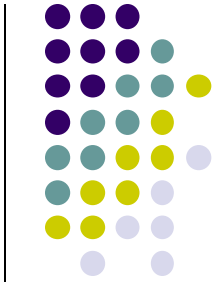
NGS Overview and Terminology

- NGS workflow overview
- Sequencing technology & terminology
- Experiment types & library complexity
- Sequence duplication



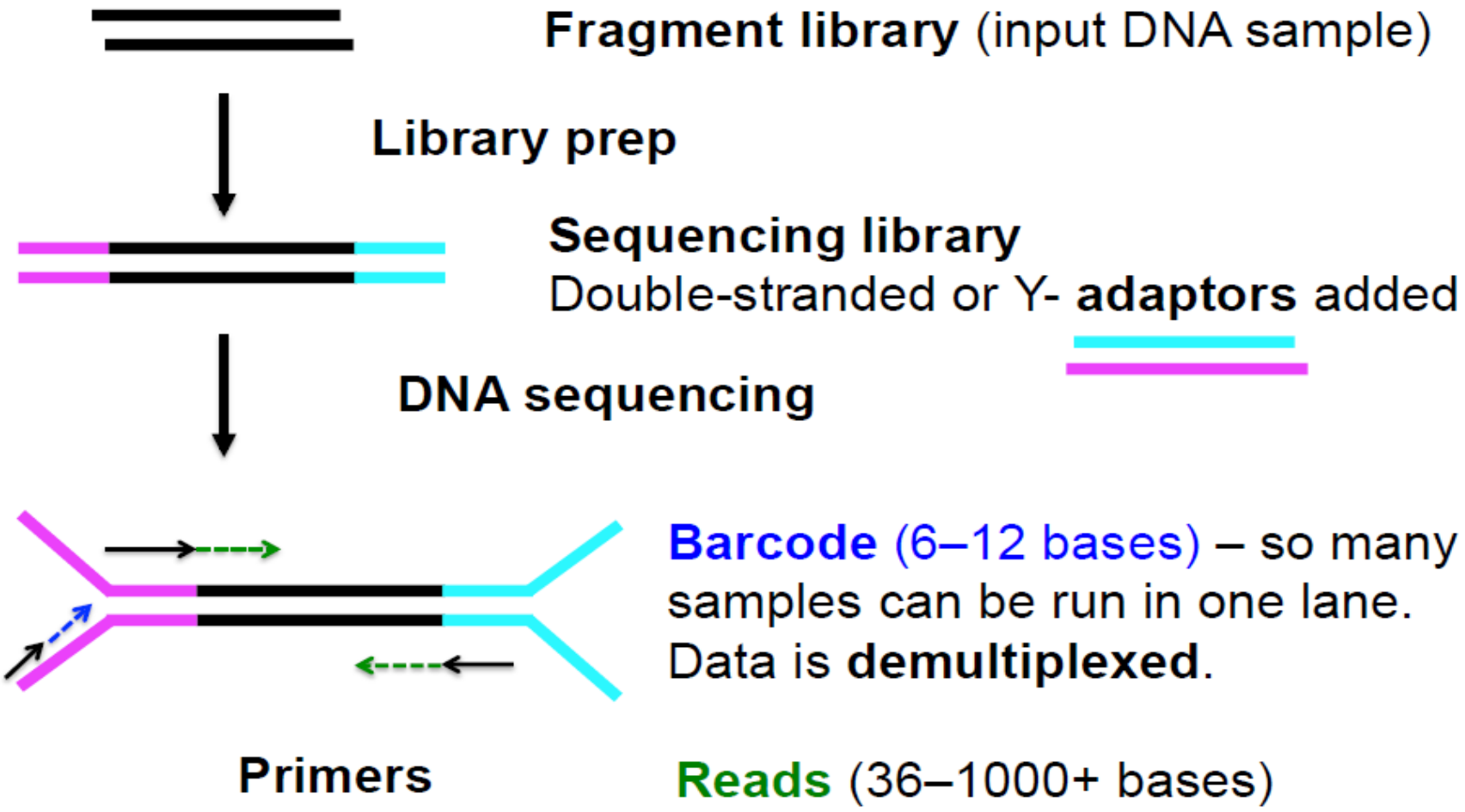

core processes
NGS Workflow
upstream processes
fastq
QC raw read sequences
no
assembly (genome or transcriptome)
experimental design
has reference?
yes
map reads to reference
reference genome
fasta
DNA/RNA isolation
metrics & QC
alignment metrics & QC
BAM
library preparation
basic analysis (e.g. coverage, genes)
bed, gff, vcf, etc.
next-gen sequencing
further analysis & significance determination (e.g. FPKM, peak or variant calls)
downstream processes
differential analysis
annotation
delivery of raw reads
confident calls
motif analysis
custom analysis

Sequencing technologies



- Illumina (Solexa) now dominant
  - Official Illumina video
  - Another Illumina video
  - Broad Center GA Boot Camp
- PacBio, up and coming (?)
  - Single Molecule Real-Time Sequencing (SMRT)
  - Much longer reads than Illumina (7kB+)
- Many others
  - Comparison of NGS technologies (Liu et al., 2012)
    http://www.hindawi.com/journals/bmri/2012/251364/

Read sequence terminology



- Adapter areas include primers, barcode
  - sequencing facility will have more information

https://wikis.utexas.edu/display/GSAF/Illumina+-+all+flavors

Types of Illumina sequencing

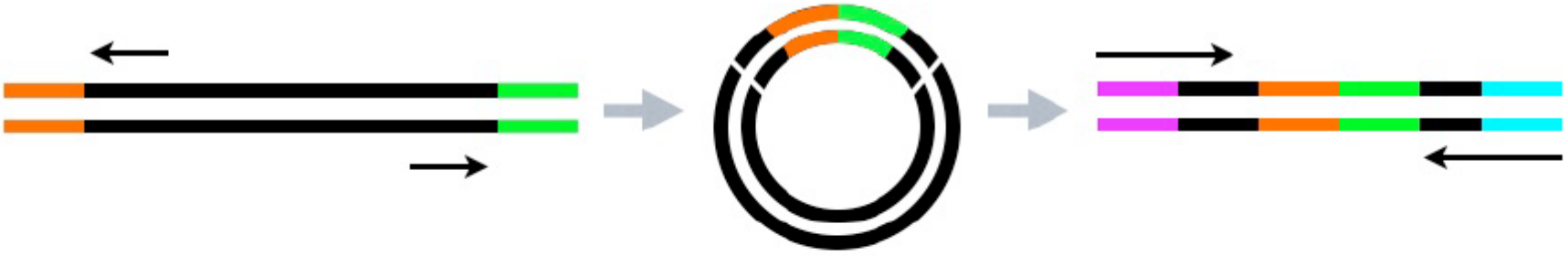
single-end

independent reads

paired-end

two inwardly oriented reads separated by ~200 nt

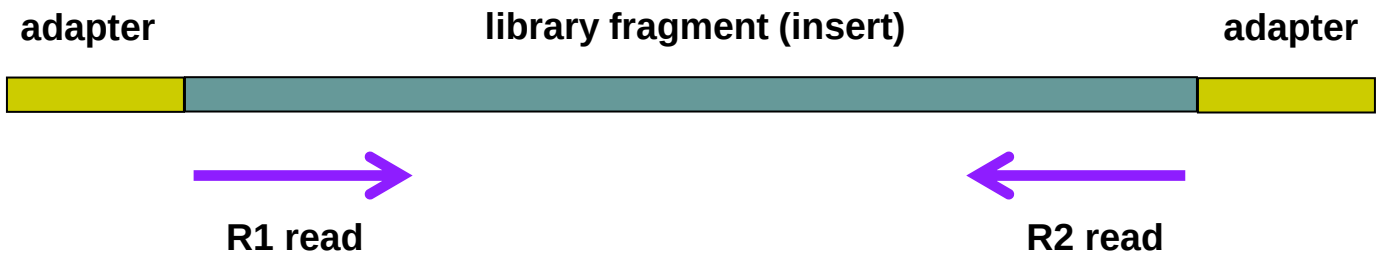
mate-paired



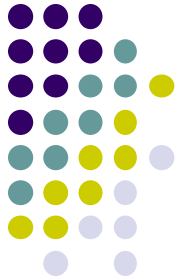
two outwardly oriented reads separated by ~3000 nt

Reads and Fragments

- With paired-end sequencing, keep in mind the distinction between
  - the library ***fragment*** that was sequenced
    - also called ***inserts***
  - the ***sequence reads*** (R1 & R2) you receive
    - also called ***tags***
- There is considerable confusion of terminology in this area!
  - Be sure you request depth in ***read pairs*** for paired-end sequencing

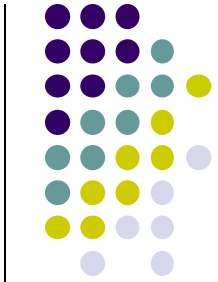


Single end vs Paired end



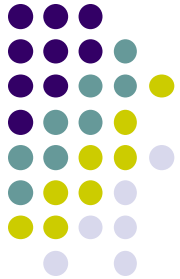
- ***paired end*** (PE) reads can be mapped more reliably
  - especially against lower complexity genomic regions
    - when one member of a read pair does not align well, it can still be "rescued" if its mate maps well
  - they also provide more bases around a locus
    - e.g. for analysis of polymorphisms
  - actual fragment sizes can be determined
    - from the alignment records for each dual-mapping "proper pair"
  - they also help distinguish the true complexity of a library
    - by clarifying which *fragments* are duplicates
- ***but*** PE reads are more expensive – and larger
  - more storage space and processing time required

Some Experiment Types



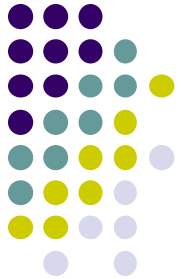
- Genomic sequencing
  - ***library***: all genomic DNA
  - **applications**: genome assembly; genomic variation analysis
- Exome sequencing
  - ***library***: DNA from (subset of) exonic regions (use special kits)
  - ***applications***: polymorphism/SNP detection; genotyping
- RNA-seq
  - ***library***: extract RNA converted to cDNA
    - all fragments → total or mRNA; small fragments → miRNA
  - ***applications***: differential gene expression; isoform discovery
- ChIP-seq
  - ***library***: DNA bound by transcription factor or histones
    - or RNA bound by specific proteins in RIP-seq
    - use target-specific antibodies to pull down after formaldehyde cross-linking
  - ***applications***: analysis of regulatory networks, dis-regulatory programs

Sequencing depth



- No single answer to how much depth is adequate
- Depends on:
  - genome size
    - prokaryotes – 1-10 Megabases
    - lower eukaryotes (fungi, algae, worms) – 10-100 Megabases
    - higher eukaryotes – 1-10 Gigabases (amphibians/plants are larger)
  - library fragment enrichment
    - e.g. ChIP-seq or RIP-seq
  - theoretical library complexity
    - genomic resquencing ***vs*** 4c
  - desired sensitivity
    - e.g. looking for rare mutations

Library complexity is primarily a function of experiment type



higher complexity

less enrichment for specific sequences

genomic

MNase-seq

exon capture

RNA-seq

ChIP-seq

miRNA-seq

more enrichment for specific sequences

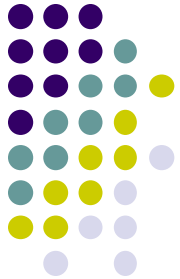
lower complexity

lower sequence duplication expected
more sequencing depth required

… and
- *sequencing depth*
- *genome size*

higher sequence duplication expected
less sequencing depth required

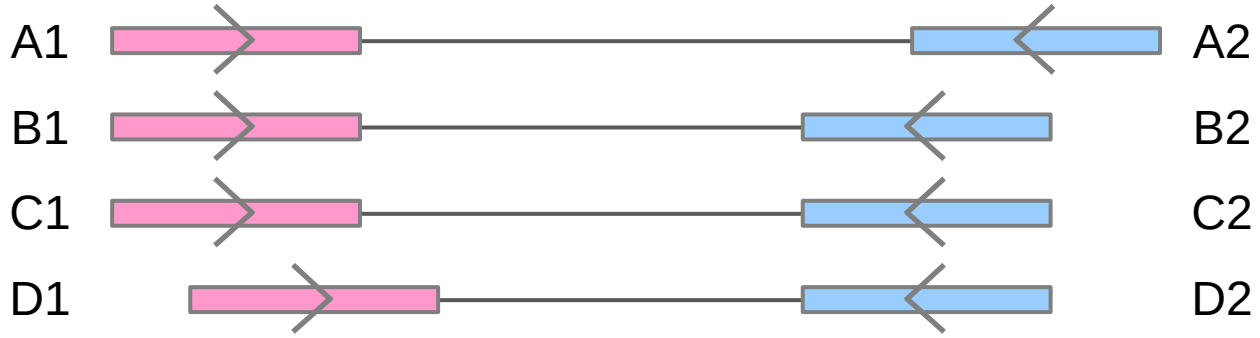
Sequence Duplication



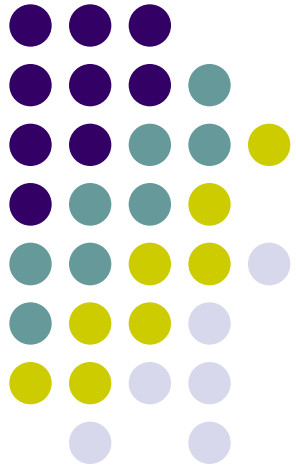
- The set of sequences you receive can contain exact duplicates
  - remember, each read comes from a different cluster, hence from a different DNA molecule
  - "high duplication" = "low complexity"
- Duplication can arise from:
  1. repeated sequencing of species enriched in your library (biological)
  2. sequencing of amplification artifacts (technical)
     - e.g. differentially enriched PCR species
  - ***cannot tell which using standard sequencing methods!***
    - new "molecular barcode" methods can address this shortcoming
- Different experiment types have different expected sequence duplication profiles
  - whole genome/exome → low duplication
  - miRNA sequencing → high duplication

Read vs Fragment duplication

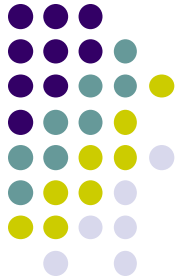
- Consider the 4 fragments below
  - 4 R1 reads (pink), 4 R2 reads (blue)
- Duplication when only 1 end considered
  - A1, B1, C1 have identical sequences, D1 different
    - 2 unique + 2 duplicates = 50% duplication rate
  - B2, C2, D2 have identical sequences, A2 different
    - 2 unique + 2 duplicates = 50% duplication rate
- Duplication when both ends considered
  - fragments B and C are duplicates (same external sequences)
    - 3 unique + 1 duplicate = 25% duplication rate



The FASTQ format



FASTQ files



- Nearly all sequencing data now delivered as FASTQ files
  - usually compressed to save space
    - (**gzip**'d, with **.gz** file extension)
  - best practice: leave them that way!
    - 3x to 6x space saving
    - most tools handle **gzip**'d FASTQ
- Paired-end sequencing data comes in 2 FASTQs
  - one each for R1 and R2 reads

    Sample_MyTubeID_L008_R1.fastq.gz

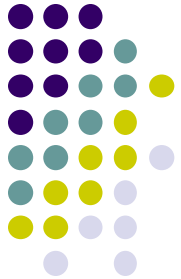
    Sample_MyTubeID_L008_R2.fastq.gz
  - ***order of reads is identical***
    - aligners rely on this "name ordering" for PE

FASTQ format

- Text format for storing sequence and quality data
  - http://en.wikipedia.org/wiki/FASTQ_format
- 4 lines per sequence:
  1. ***@read name***
  2. ***called base sequence*** **(ACGTN)**
     always 5' to 3'; usually excludes 5' adapter/barcode
  3. ***+optional read name***
  4. ***base quality scores encoded as text characters***
- FASTQ representation of a single, 50 base R1 sequence

```
@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 1:N:0:ACTTGA
ATTCTCCAAGATTTGGCAAATGATGAGTACAATTATATGCCCCAATTTACA
+
?@@?DD;?;FF?HHBB+:ABECGHDHDCF4?FGIGACFDFH;FHEIIIB9?
```

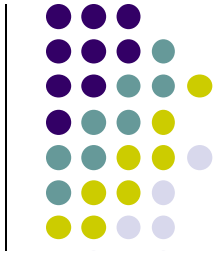
FASTQ read names



- Illumina read names encode information about the source cluster
  - unique identifier ("fragment name") begins with @, then:
    - sequencing machine name
    - lane number
    - flowcell grid coordinates
    - R1, R2 reads will have the same name
  - a space separates the name from extra read information:
    - end number (1 for R1, 2 for R2)
    - two qualtiy fields (N = *not* QC failed)
    - barcode sequence

```
@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 1:N:0:ACTTGA
@HWI-ST1097:97:D0WW0ACXX:8:1101:2007:2085 2:N:0:ACTTGA
```

FASTQ quality scores



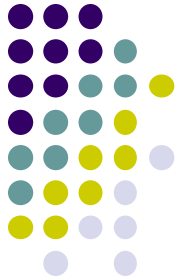
http://www.asciitable.com/

```
Quality character   !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
                    |         |         |         |         |
ASCII Value         33        43        53        63        73
Base Quality (Q)    0         10        20        30        40
```

$$\textbf{Probability of Error} = 10^{-Q/10}$$

- Base qualities expressed as ***Phred*** scores
  - log scaled, higher = better
  - 20 = $1/10^2$ = 1/100 errors, 30 = $1/10^3$ = 1/1000 errors
- In older FASTQ files, ASCII offsets may differ
  - modern Sanger format shown above
  - see http://en.wikipedia.org/wiki/FASTQ_format for others

Multiple lanes



- Sometimes the sequencing facility splits your sample across lanes
  - one submitted sample may be delivered as multiple FASTQ files

    Lane1: **Sample_MyTubeID_L001_R1.fastq.gz**, **Sample_MyTubeID_L001_R2.fastq.gz**

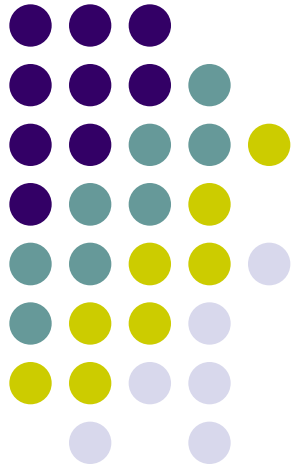
    Lane8: **Sample_MyTubeID_L008_R1.fastq.gz**, **Sample_MyTubeID_L008_R2.fastq.gz**
- Your sample may be re-run to "top off" requested read depth
  - be careful with the file names!
    - if run in the same lane, the FASTQ file names will be the same

    1st run: **Sample_MyTubeID_L003_R1.fastq.gz**

    2nd run : **Sample_MyTubeID_L003_R1.fastq.gz**
- Best practice
  - keep original data in separate directories by date & project
  - process data from multiple lanes separately for as long as possible
    - e.g., through alignment, then merge the sorted BAMs
    - identical sequences from different lanes can be considered unique (non-duplicates)

Data QC & preparation

- QC of raw sequences with **FastQC** tool
- Dealing with adapters



Raw sequence quality control

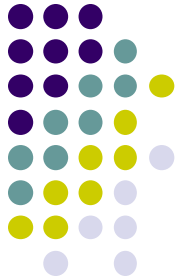
- Critical step! Garbage in = Garbage out
  - general sequence quality
    - base quality distributions
    - sequence duplication rate
  - trim 3' adapter sequences?
    - important for RNAseq
  - trim 3' bases with poor quality?
    - important for *de novo* assembly
  - other contaminants?
    - technical – samples sequenced on other lanes
    - biological – rRNA in RNAseq
- Know your data
  - sequencing center pre-processing
    - 5' barcode removal; QC-failed reads filtered
  - PE reads? relative orientations?
  - technology specific issues?
    - e.g. PAR clip should produce C→T transitions

3’ Adapter contamination

A. reads short compared to fragment size (no contamination)

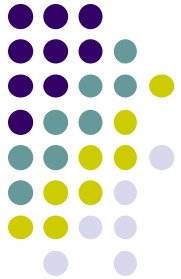
B. reads long compared to library fragment (3’ adapter contamination)

FastQC



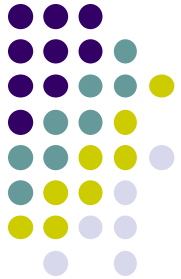
- Quality Assurance tool for FASTQ sequences
- Can run as interactive tool or command line
- Input:
  - FASTQ file(s)
  - run on both R1, R2 files
- Output:
  - directory with html & text reports
    - **fastqc_report.html**
    - **fastqc_data.txt**

FastQC resources



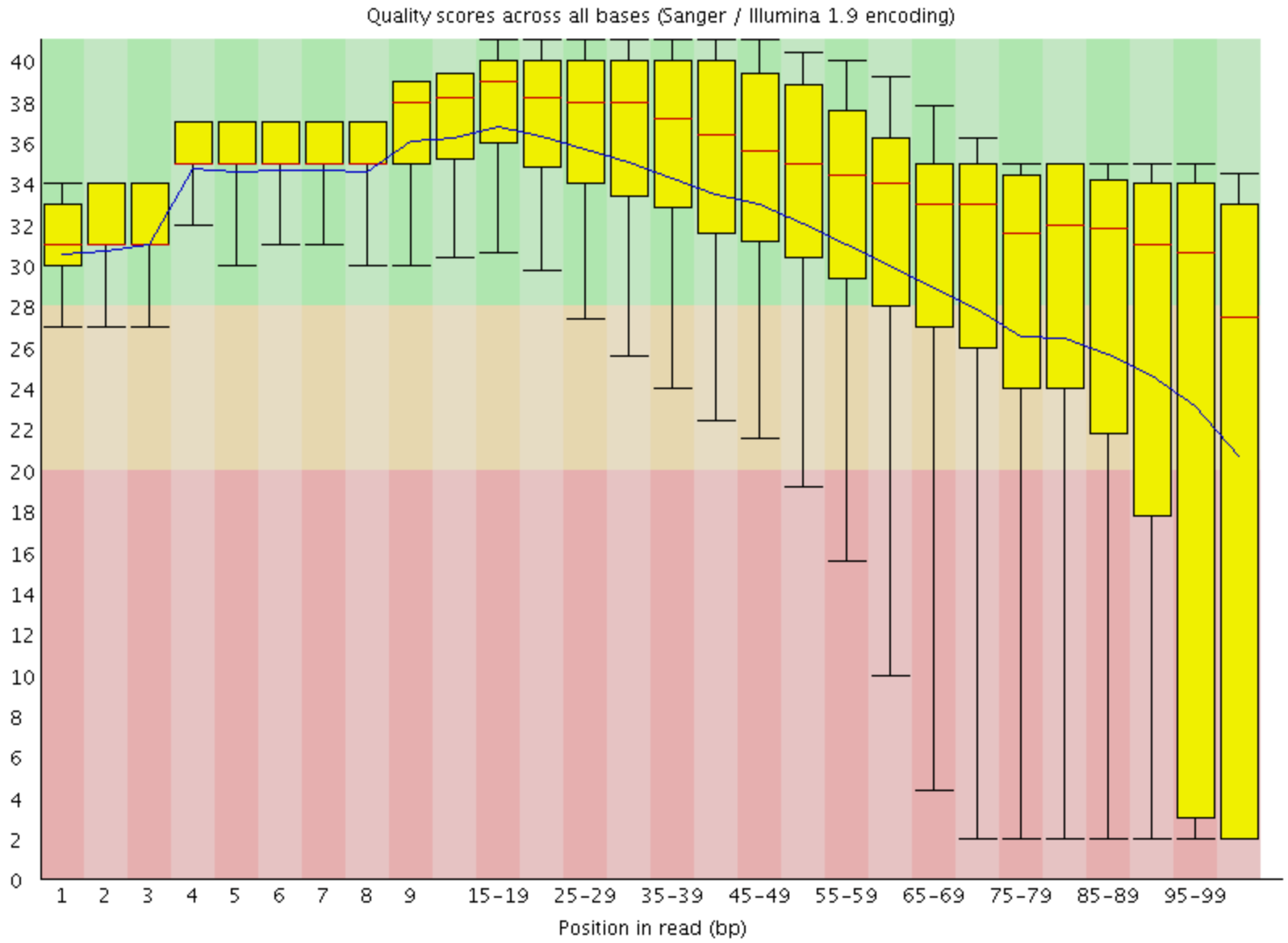
- FastQC website:
  http://www.bioinformatics.babraham.ac.uk
- FastQC report documentation:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/
- Good Illumina dataset:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc.html
- Bad Illumina dataset:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html
- Real Yeast ChIP-seq dataset:
  http://web.corral.tacc.utexas.edu/BioITeam/yeast_stuff/Sample_Yeast_L005_R1.cat_fastqc/fastqc_report.html

Most useful FastQC reports

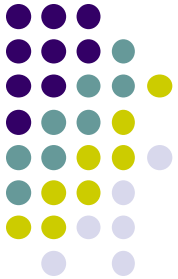


- Should I trim low quality bases?
  - *Per-base sequence quality Report*
    - based on *all* sequences
- Do I need to remove adapter sequences?
  - *Overrepresented sequences Report*
    - based on *1st 200,000* sequences
- How complex is my library?
  - *Sequence duplication levels Report*
    - estimate based on *1st 200,000* sequences

FastQC Per-base sequence quality report



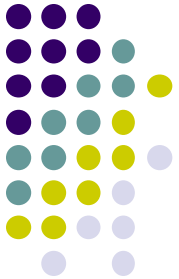
FastQC Overrepresented sequences report



- **FastQC** knows Illumina adapter sequences
- Here ~9-10% of sequences contain adapters
  - calls for trimming

Sequence	Count	Percentage	Possible Source
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATG	60030	5.01369306977828	TruSeq Adapter, Index 1 (97% over 37bp)
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATGC	42955	3.5875926338884896	TruSeq Adapter, Index 1 (97% over 37bp)
CACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATGCCGTCTTCTGCT	3574	0.29849973398946483	RNA PCR Primer, Index 40 (100% over 41bp)
CAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTAT	2519	0.2103863542024236	TruSeq Adapter, Index 1 (97% over 37bp)
GAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTAT	1251	0.10448325887543942	TruSeq Adapter, Index 1 (97% over 37bp)

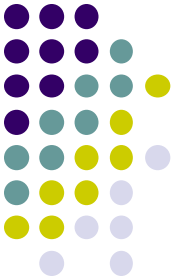
Overrepresented sequences



- Here < 1% of sequences contain adapters
  - trimming optional

Sequence	Count	Percentage	Possible Source
AACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTA	102020	1.0707851766890004	No Hit
AATTCTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCA	89437	0.9387160737848865	No Hit
AAAGGATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTCCGAACCCGT	89427	0.9386111154260659	No Hit
TACCTGGTTGATCCTGCCAGTAGTCATATGCTTGTCTCAAAGATTAAGCC	87604	0.9194772066130483	No Hit
ATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTCCGAACCCGTCGGCT	65829	0.6909303802809273	No Hit
TCTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCATTT	65212	0.6844544495416888	No Hit
TAAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAAC	61582	0.646354565289767	No Hit
CTCGGATAACCGTAGTAATTCTAGAGCTAATACGTGCAACAAACCCCGAC	59180	0.6211435675010296	No Hit
ATGGATCCGTAACTTCGGGAAAAGGATTGGCTCTGAGGGCTGGGCTCGGG	56982	0.598073720232235	No Hit
AAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACG	54813	0.5753082522040206	No Hit
CTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCATTTA	40019	0.4200328561646452	No Hit
AGAACTCCGCAGTTAAGCGTGCTTGGGCGAGAGTAGTACTAGGATGGGTG	39753	0.4172409638200141	No Hit
ACTCGGATAACCGTAGTAATTCTAGAGCTAATACGTGCAACAAACCCCGA	38867	0.4079416532284981	No Hit
ACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTAG	38438	0.40343893963508914	No Hit
ACTTCGGGAAAAGGATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTC	37406	0.3926072370047907	No Hit
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACTGACCAATCTCGTATG	34199	0.35894709133098535	TruSeq Adapter, Index 4 (100% over 49bp)
GAACCTTGGGATGGGTCGGCCGGTCCGCCTTTGGTGTGCATTGGTCGGCT	34099	0.3578975077427782	No Hit

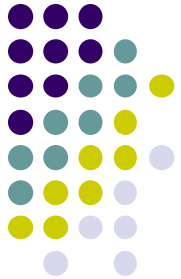
Overrepresented sequences



- Here nearly 1/3 of sequences some type of non-adapter contamination
  - **BLAST** the sequence to identify it

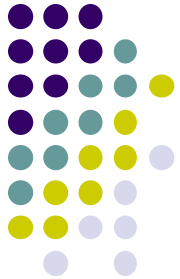
Sequence	Count	Percentage	Possible Source
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGTGTCAAGTGG	5632816	32.03026785752871	No Hit
TATTCTGGTGTCCTAGGCGTAGAGGAACAACACCAATCCATCCCGAACTT	494014	2.8091456822607364	No Hit
TCAAACGAGGAAAGGCTTACGGTGGATACCTAGGCACCCAGAGACGAGGA	446641	2.539765344040083	No Hit
TAAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAAC	179252	1.0192929387357474	No Hit
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGGGTCAAGTGG	171681	0.9762414422996221	No Hit
AACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTA	143415	0.8155105483274229	No Hit
AGAACATGAAACCGTAAGCTCCCAAGCAGTGGGAGGAGCCCTGGGCTCTG	111584	0.6345077504066322	No Hit
AAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACG	111255	0.6326369351474214	No Hit
ATTACGATAGGTGTCAAGTGGAAGTGCAGTGATGTATGCAGCTGAGGCAT	73682	0.41898300890326096	No Hit
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGTGTCAAGGGG	71661	0.4074908580252516	No Hit
GGATGCGATCATACCAGCACTAATGCACCGGATCCCATCAGAACTCCGCA	69548	0.3954755612388914	No Hit
ATATTCTGGTGTCCTAGGCGTAGAGGAACAACACCAATCCATCCCGAACT	54017	0.30716057099328803	No Hit

Dealing with adapters



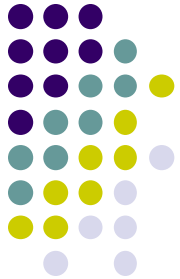
- Three main options:
  1. Hard trim all sequences by specific amount
  2. Remove adapters specifically
  3. Peform a local (vs global) alignment

Hard trim by specific length



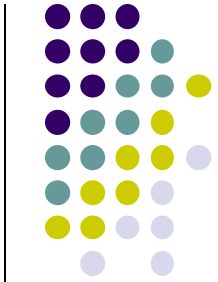
- E.g. trim 100 base reads to 50 bases
- ***Pro:***
  - Can eliminate vast majority of adapter contamination
  - Fast, easy to perform
  - Low quality 3’ bases also removed
- ***Con:***
  - Removes information you may want
    - e.g. splice junctions for RNAseq, coverage for mutation analysis
  - Not suitable for very short library fragments
    - e.g. miRNA libraries

Trim adapters specifically



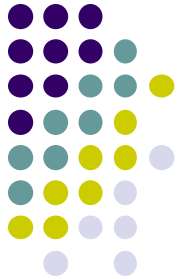
- ***Pro:***
  - Can eliminate vast majority of adapter contamination
  - Minimal loss of sequence information
    - still ambiguous: are 3'-most bases part of sequence or adapter?
- ***Con:***
  - Requires knowledge of insert fragment structure and adapters
  - Slower process; more complex to perform
  - Results in heterogenous pool of sequence lengths
    - can confuse some tools (rare)

FASTQ trimming



- Tools:
  - **cutadapt** – https://cutadapt.readthedocs.org/en/stable
  - **trimmomatic** – http://www.usadellab.org/cms/?page=trimmomatic
  - **FASTX-toolkit** – http://hannonlab.cshl.edu/fastx_toolkit/
- Features:
  - hard-trim specific number of bases
  - trimming of low quality bases
  - specific trimming of adapters
  - support for trimming paired end read sets
    - typically reads less than a specified length *after trimming* are discarded
    - leads to different sets of R1 and R2 reads unless care taken
      - aligners do not like this!
  - ability to “collapse” duplicate reads into a single read plus count
    - important for applications that require *very* deep sequencing
      - e.g. genome/transcriptome assembly

Local vs. global alignment



- ***Global***
  - requires query sequence to map ***fully*** (end-to-end) to reference
- ***Local***
  - allows a ***subset*** of the query sequence to map to reference

global *(end-to-end) alignment of query*

CACAAGTACAATTATACAC

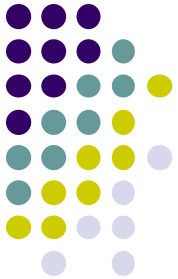
local *(subsequence) alignment of query*

CTAGCTTATCGCCCTGAAGGACT

TACATACACAAGTACAATTATACACAGACATTAGTTCTTATCGCCCTGAAAATTCTCC

reference sequence

Perform local alignment



- ***Pro:***
  - mitigates adapter contamination while retaining full query sequence
  - minimal ambiguity
    - still ambiguous: are 5'/3'-most bases part of sequence or adapter?
- ***Con:***
  - not supported by many aligners
    - e.g. not by the **tophat** splice-aware aligner for RNAseq
  - slower alignment process
  - more complex post-alignment processing may be required
- Aligners with local alignment support:
  - **bwa mem**
  - **bowtie2 --local**

FastQC Sequence duplication report

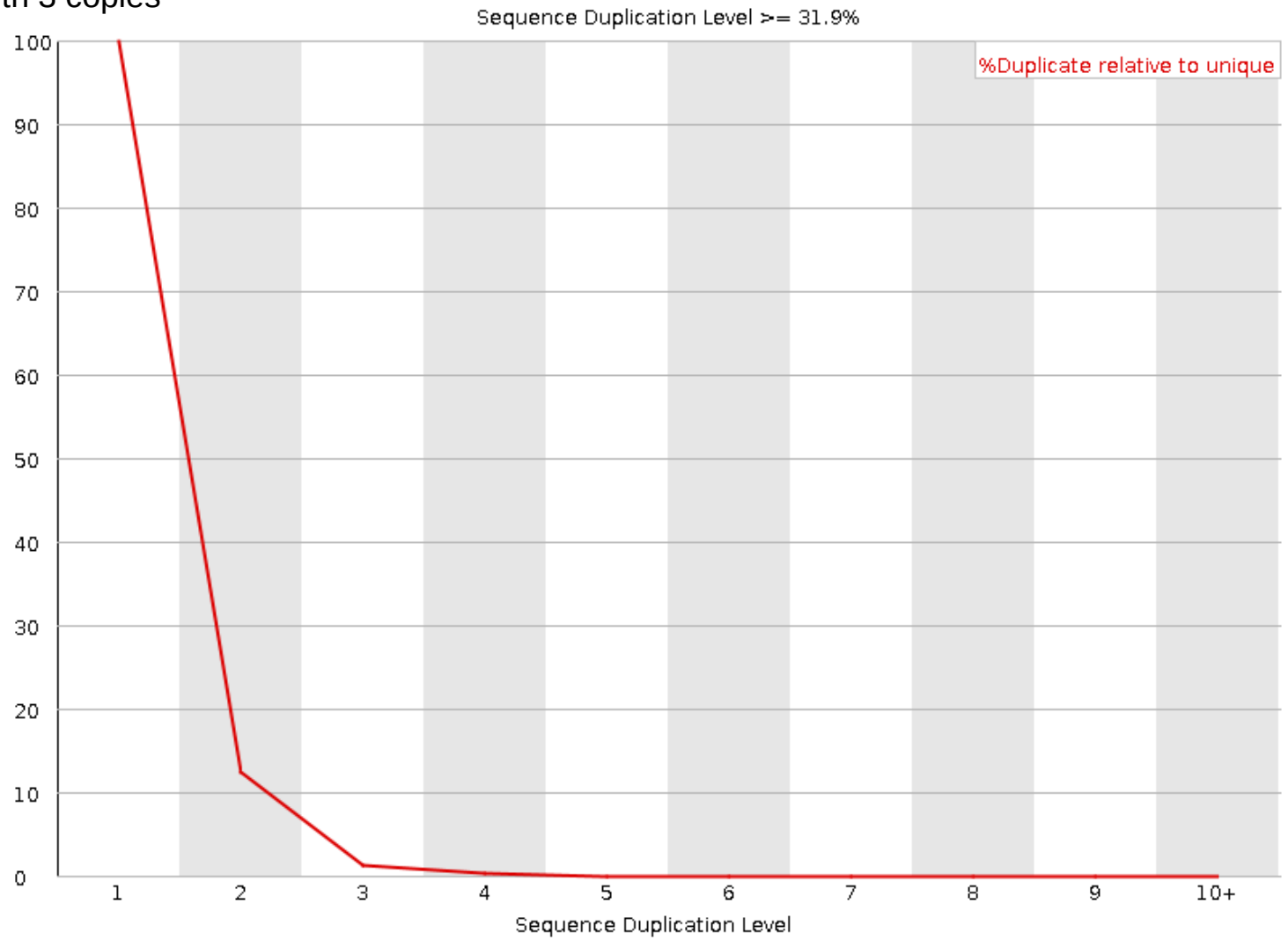
Yeast ChIP-seq

for every 100 unique sequences
there are:

~12 sequences w/2 copies

~1-2 with 3 copies

some duplication expected due to IP enrichment



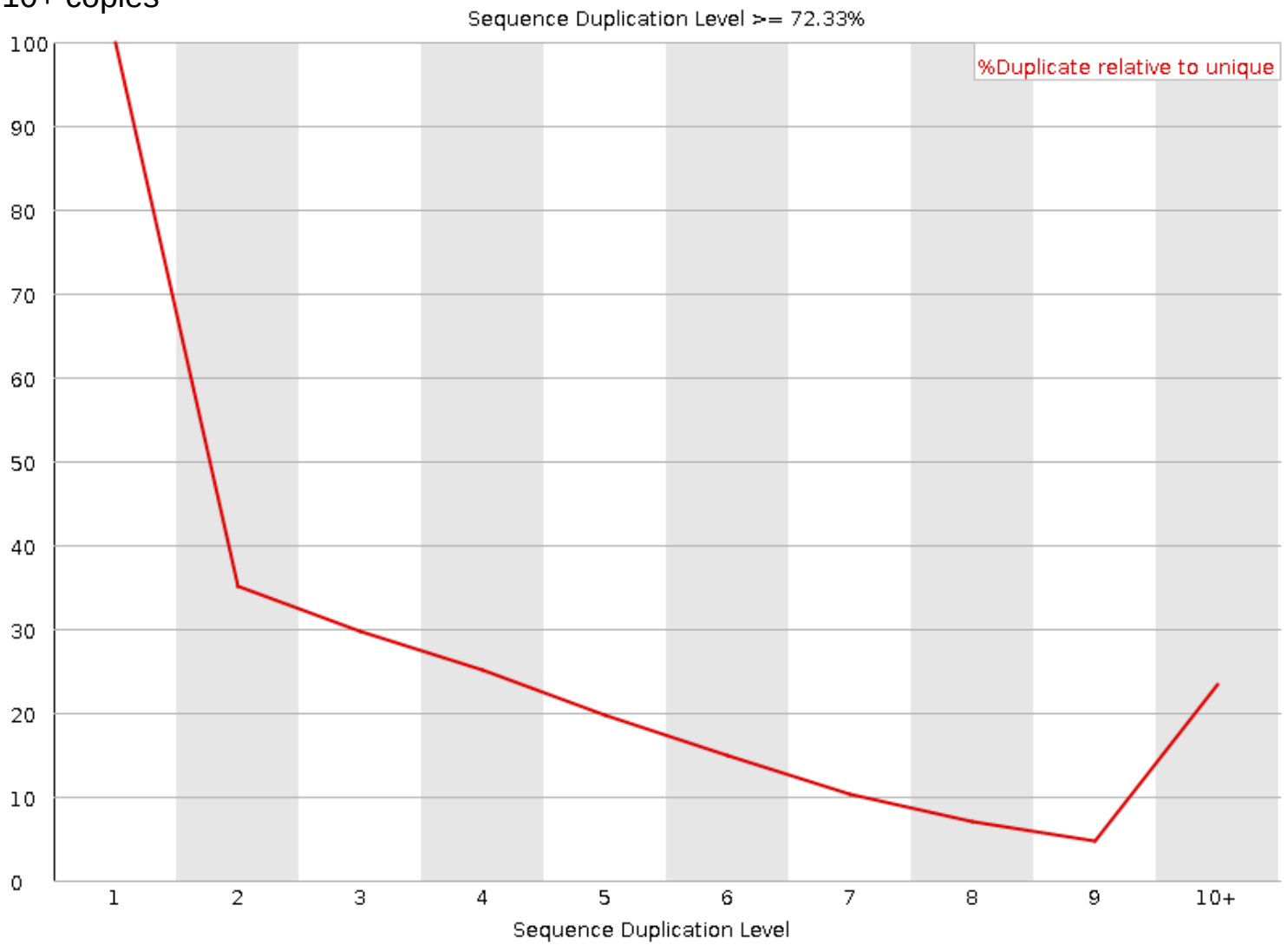
Sequence duplication report

Yeast ChIP-exo

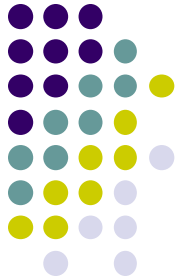
for every 100 unique sequences there are:

- ~35 sequences w/2 copies
- ~22 with 10+ copies

success! protocol expected to have high duplication



Library complexity is primarily a function of experiment type



higher complexity

less enrichment for specific sequences

genomic

MNase-seq

exon capture

RNA-seq

ChIP-seq

miRNA-seq

more enrichment for specific sequences

lower complexity

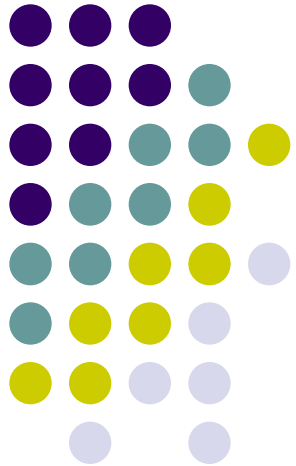
lower sequence duplication expected
more sequencing depth required

… and
- *sequencing depth*
- *genome size*

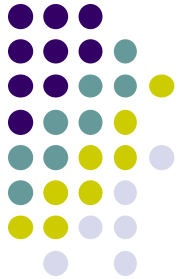
higher sequence duplication expected
less sequencing depth required

Alignment to a reference genome

- Alignment overview & concepts
- Preparing a reference genome
- Alignment steps

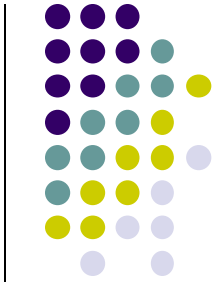


Short Read Aligners



- Short read mappers determine the placement of query sequences against a known reference
  - **BLAST**:
    - one query sequence (or a few)
    - many matches for each
  - short read aligners
    - many millions of query sequences
    - want only one “best” mapping (or a few)
  - many such aligners available
    http://en.wikipedia.org/wiki/List_of_sequence_alignment_software
- We use 2 of the most popular
  - **bwa** (Burrows Wheeler Aligner) by Heng Li
    http://bio-bwa.sourceforge.net/
  - **bowtie2** – part of the Johns Hopkins Tuxedo suite of tools
    http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

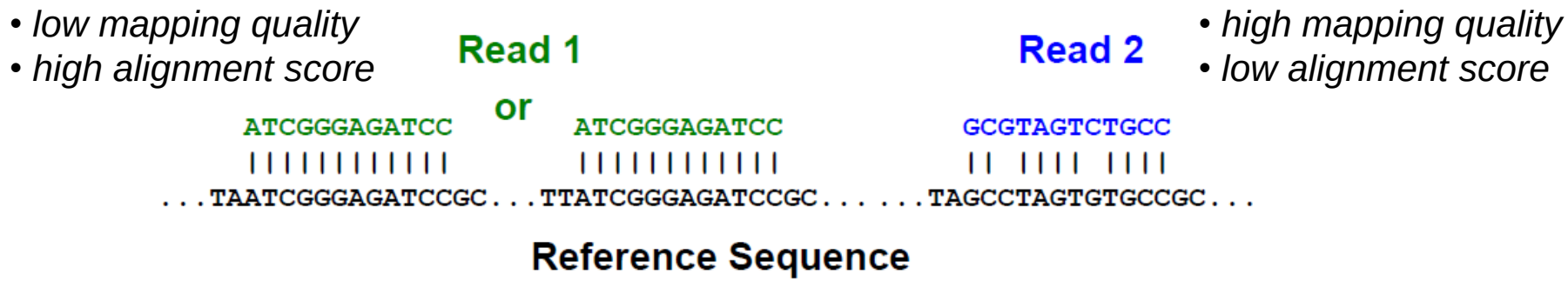
Aligner criteria



- Adoption and currency
  - widespread use by bioinformatics community
  - still being actively developed
- Features
  - well understood algorithm(s)
  - support for a variety of input formats and read lengths
  - detection of indels and gaps
  - makes use of base qualities
  - handling of multiple matches?
- Usability
  - configurability and transparency of options
  - ease of installation and use
- Resource requirements
  - speed (“fast enough”)
  - scalability (takes advantage of multiple processors)
  - reasonable memory footprint

Mapping vs Alignment

- ***Mapping*** determines one or more "seed" positions (a.k.a "hits") where a read shares a subsequence with the reference
- ***Alignment*** starts with the seed and determines how read bases are best matched, base-by-base, around the seed
- Mapping quality and alignment scores are both reported
- High ***mapping quality*** ≠ High ***alignment score***
  - ***mapping quality*** describes ***positioning / sequence uniqueness***
    - indicates the complexity/information content of the sequence ("mappability")
    - reflects the probability that the read is *incorrectly* mapped to the reported location
    - is a Phred score: $P(\text{mis-mapped}) = 10^{-\text{mappingQuality}/10}$
  - ***alignment score*** describes ***fit / match***
    - reflects the correspondence between the read and the reference sequences



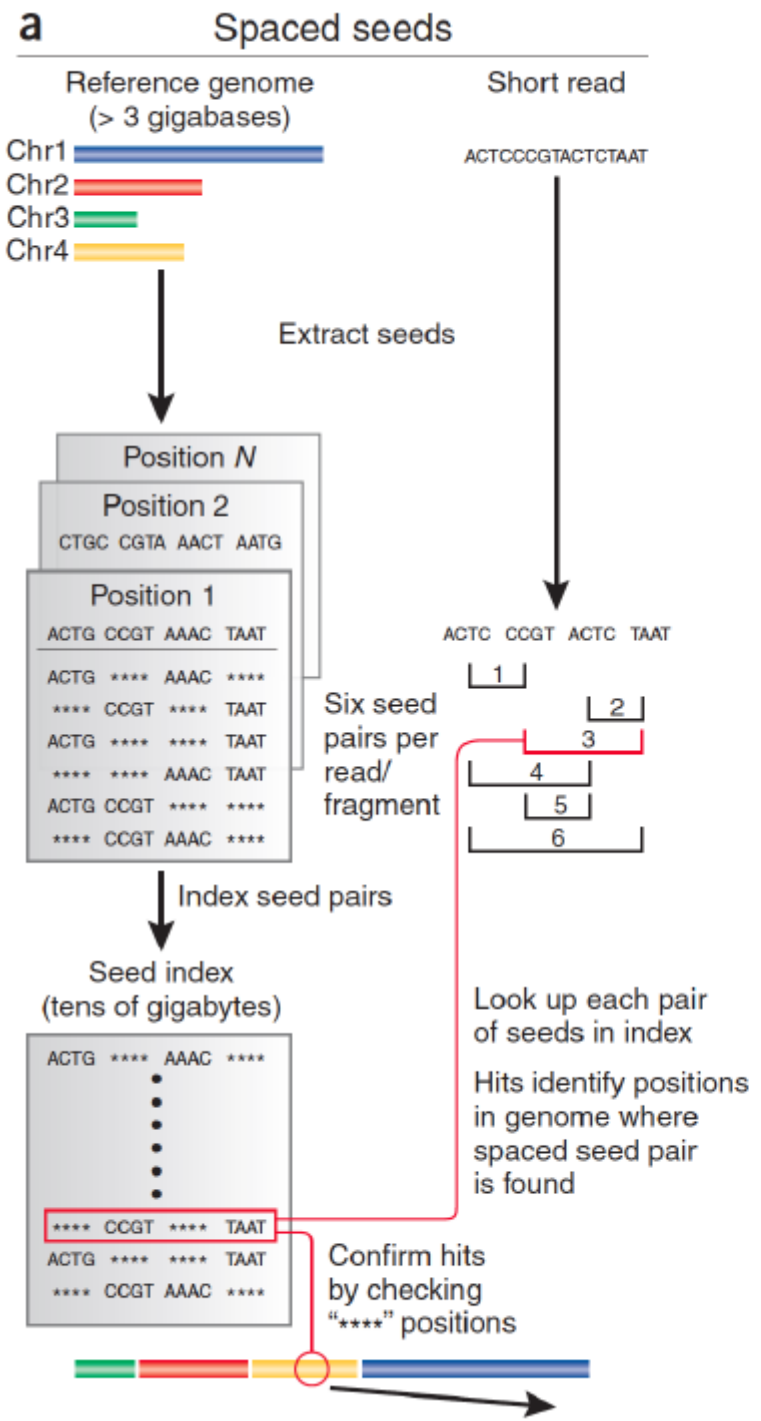
Some Aligners

Two main mapping algorithms: *spaced seeds*, *suffix-array tries*

	Algorithm	Gapped	Quality-aware	Colorspace aware
BLAST	Hash table	Y	N	N
BLAT/SSHA2	Hash table	N	N	N
MAQ	Spaced seed	N	N	N
RMAP	Spaced seed	N	Y	N
ZOOM	Spaced seed	N	--	N
SOAP	Spaced seed	N	N	N
Eland	Spaced seed	N	N	N
SHRiMP	Q-gram/multi-seed	Y	Y	Y
BFAST	Q-gram/multi-seed	Y	Y	Y
Novoalign	Multi-seed + Vectorized SW	Y	Y	Y
clcBio	Multi-seed + Vectorized SW	Y	Y	Y
MUMmer	Tries	Y	N	N
OASIS	Tries	Y	--	--
VMATCH	Tries	Y	--	--
BWA/BWA-SW	Tries	Y	Y	Y
BOWTIE	Tries	Y	Y	Y
SOAP2	Tries	Y	N	N
Saruman	Exact (GPU)	Y	--	N

courtesy of Matt Vaughn, TACC

trie = tree structure for fast text retrieval.



Hash table enables lookup of exact matches.

Subsequence	Reference Positions
ATAGCTAATCCAAA	2341, 2617264
ATAGCTAATCCAAT	
ATAGCTAATCCAAC	134, 13311, 732661,
ATAGCTATCCAAAG	
ATAGCTAATCCATA	
ATAGCTAATCCATT	3452
ATAGCTAATCCATC	
ATAGCTATCCAATG	234456673

Table is sorted and complete so you can jump immediately to matches. (But this can take a lot of memory.)

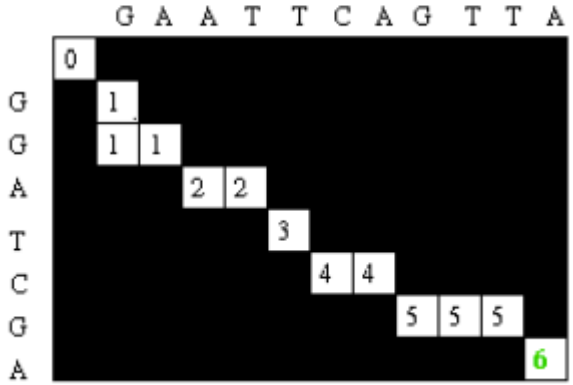
May include N bases, skip positions, etc.

Trapnell, C. & Salzberg, S. L. How to map billions of short reads onto genomes. *Nature Biotech.* **27**, 455–457 (2009).

Alignment via dynamic programming

- Dynamic programming algorithm (Smith-Waterman | Needleman-Wunsch)

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6



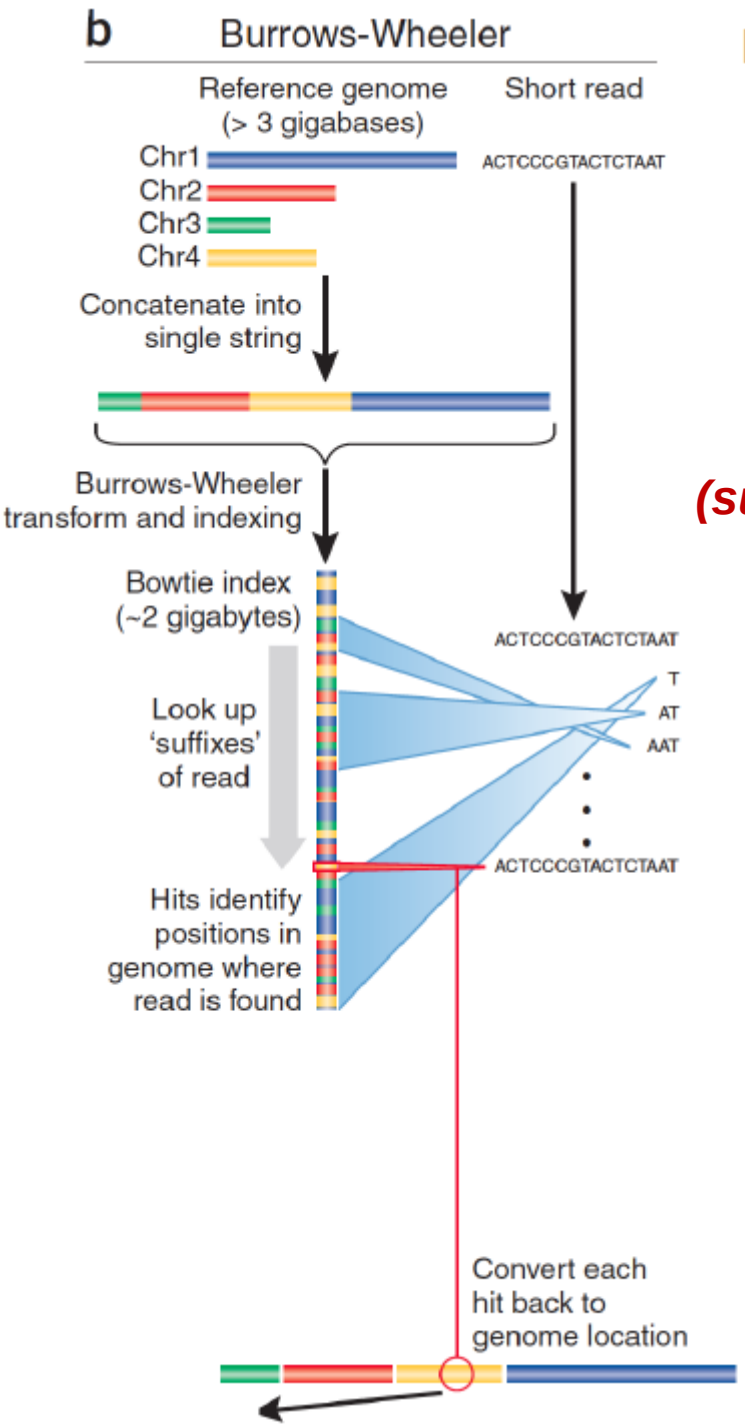

```
G _ A A T T C A G T T A
|   |   | |   |     |
G G _ A _ T C _ G _ _ A
```

- **Alignment score** = Σ
  - match reward
  - base mismatch penalty
  - gap open penalty
  - gap extension penalty
  - rewards and penalties may be adjusted for quality scores of bases involved

Reference sequence

```
ATTTGCGATCGGATGAAGACGAA
|||||||||||||||
ATTTGCGATCGGATGTTGACTTT

ATTTGCGATCGGATGAAGACG..AA
|||||||||||||||XX|||Xii||
ATTTGCGATCGGATGTTGACTTTAA
```

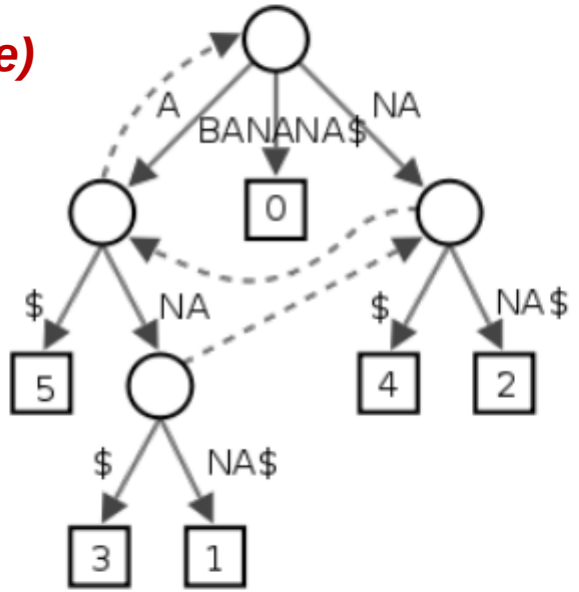


Burrows-Wheeler transform compresses sequence.

Input	SIX.MIXED.PIXIES.SIFT.SIXTY.PIXIE.DUST.BOXES
Output	TEXYDST.E.IXIXIXXSSMPPS.B..E.S.EUSFXDIIOIIIT

Suffix tree enables fast lookup of subsequences.

(suffix array trie)

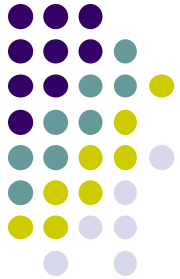


http://en.wikipedia.org/wiki/Suffix_tree

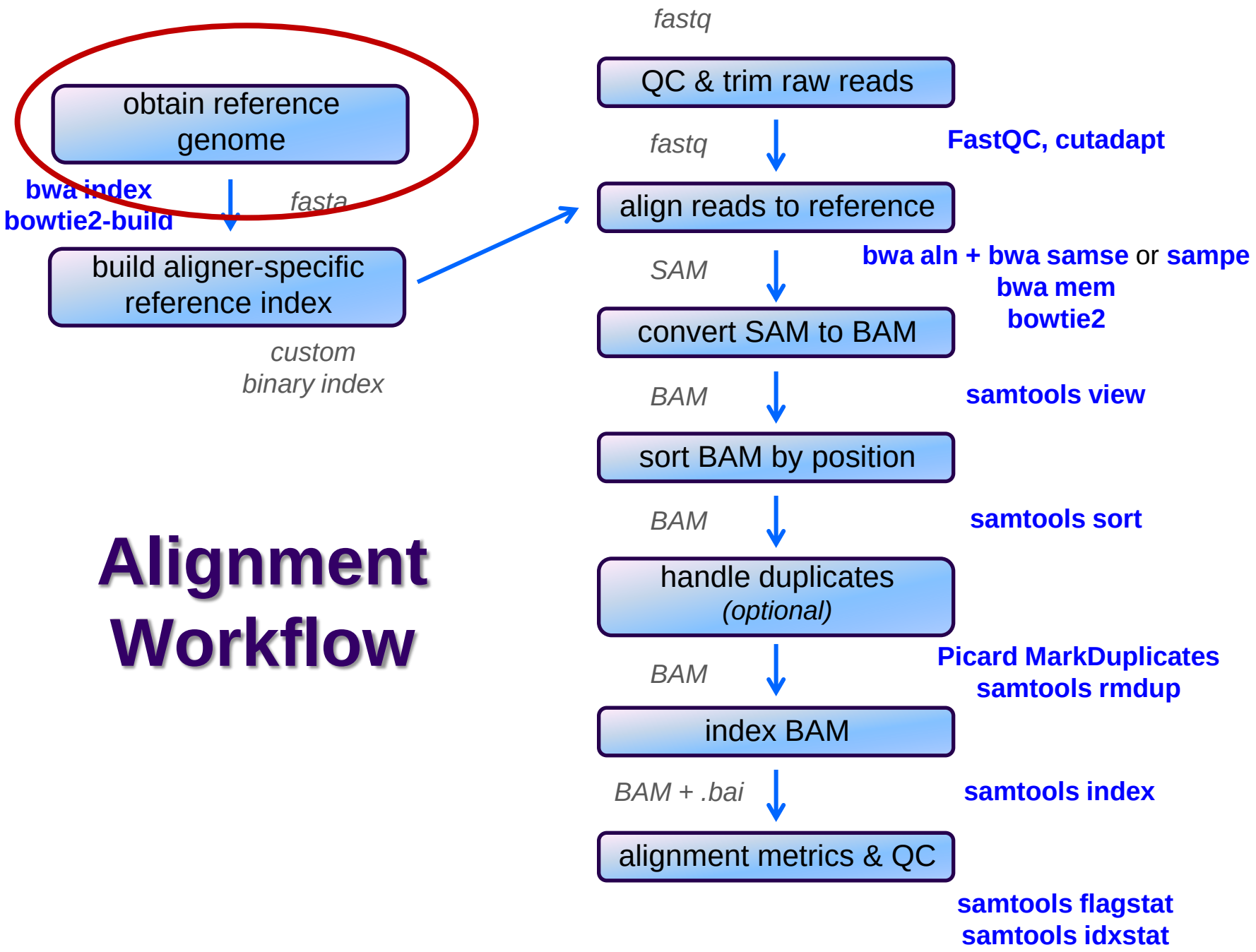
Exact matches at all positions below a node.

Trapnell, C. & Salzberg, S. L. How to map billions of short reads onto genomes. *Nature Biotech.* **27**, 455–457 (2009).

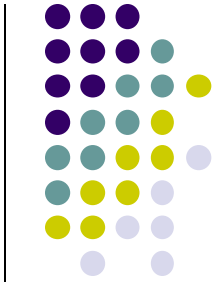
Paired End mapping



- Having paired-end reads improves mapping
  - mapping one read with high confidence anchors the pair
    - even when its mate read alone maps several places equally
- There is an expected insert size distribution based on the DNA fragment library
  - only one of a pair might map ***(singleton/orphan)***
  - both reads can map within the most likely distance range ***(proper pair)***
  - both reads can map but with an unexpected insert size or orientation ***(discordant pair)***
- The insert size is reported in the alignment record for both proper and discordant pairs

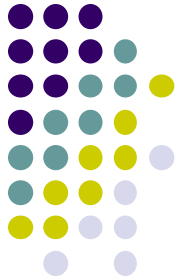

obtain reference genome
bwa index
bowtie2-build
fasta
build aligner-specific reference index
custom binary index
Alignment Workflow
fastq
QC & trim raw reads
fastq
FastQC, cutadapt
align reads to reference
SAM
bwa aln + bwa samse or sampe
bwa mem
bowtie2
convert SAM to BAM
BAM
samtools view
sort BAM by position
BAM
samtools sort
handle duplicates
(optional)
BAM
Picard MarkDuplicates
samtools rmdup
index BAM
BAM + .bai
samtools index
alignment metrics & QC
samtools flagstat
samtools idxstat

Obtaining a reference



- What is a reference?
  - any set of named sequences
    - e.g. names are chromosome names
    - technically referred to as “contigs”
- Assembled genomes
  - Ensembl, UCSC for eukaryotes
    - FASTA files (**.fa, .fasta**)
  - GenBank, NCBI for prokaryotes/microbes
    - records contain both fasta sequences and annotations
- Any set of sequences of interest, e.g:
  - transcriptome (set of gene sequences)
  - rRNA/tRNA genes (for filtering)
  - miRNA hairpin sequences from miRBase

FASTA format



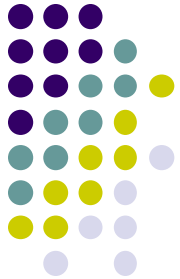
- FASTA files contain a set of sequence records
  - sequence name line
    - always starts with **>**
      - followed by name and other (optional) descriptive information
  - one or more sequence line(s)
    - never starts with **>**
- Mitochondrial chromosome sequence, human hg19

```
>chrM
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCAT
TTGGTATTTTCGTCTGGGGGGTGTGCACGCGATAGCATTGCGAGACGCTG
GAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTGCCTCATT ...
```

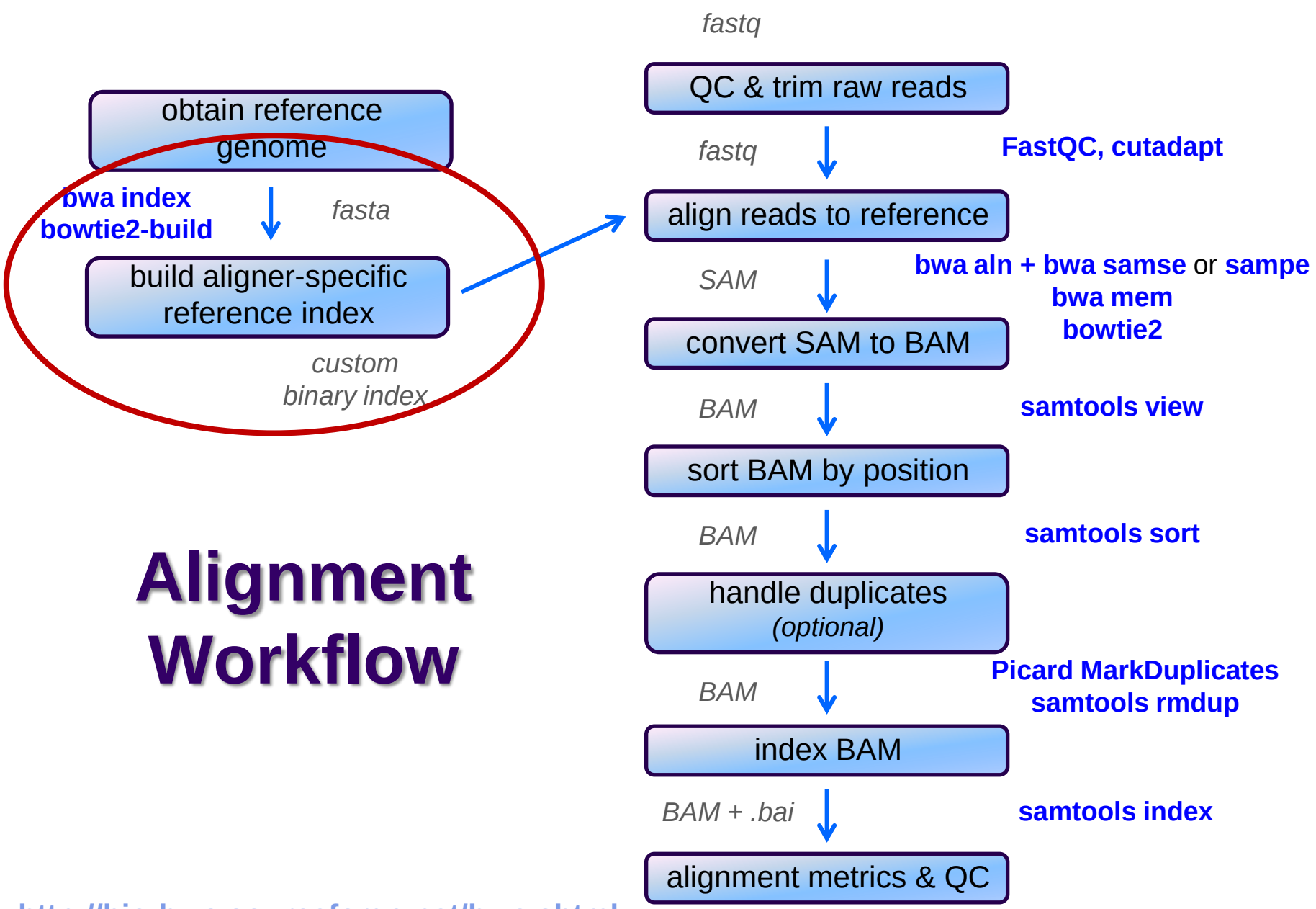
- Let-7e miRNA, human miRBase v21

```
>hsa-let-7e MI0000066 Homo sapiens let-7e stem-loop
CCCGGGCTGAGGTAGGAGGTTGTATAGTTGAGGAGGACACCCAAGGAGATCACTATACGG
CCTCCTAGCTTTCCCCAGG
```

Reference considerations



- Is it appropriate to your study?
  - close enough to your species? complete?
- Does it contain repeats? What kinds?
  - know this up front or you will be confused
- From which source? And which version?
  - UCSC hg19 vs. Ensembl GRCh37
- What annotations exist?
  - references lacking feature annotations are much more challenging
- Watch out for sequence name issues!
  - sequence names are different between UCSC/Ensembl
    - e.g. “chr12” vs. “12”
  - ***annotation sequence names must match names in your reference!***
  - long sequence names can cause problems
    - rename: `>hsa-let-7e_MI0000066_Homo_sapiens_let-7e stem-loop`
    - to: `>hsa-let-7e`

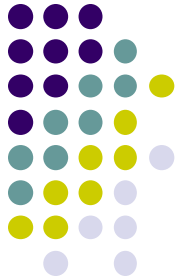


Alignment Workflow

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

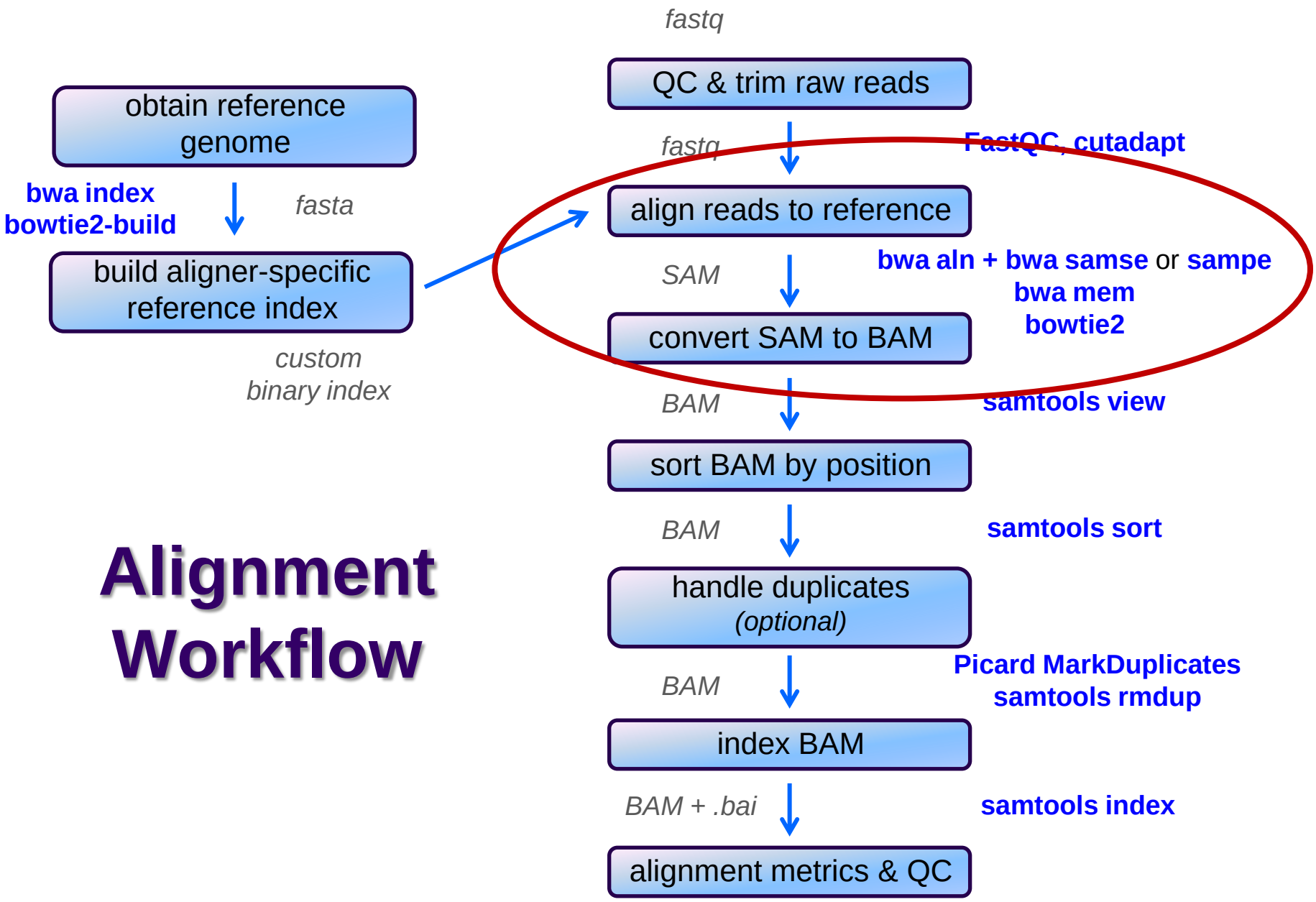
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

Building a reference index



- Index format is specific to each aligner
  - may take several hours to build
    - but you build each index once, use for multiple alignments
- Input:
  - a FASTA file
- Output:
  - a number of binary files the aligner will use
- Best practice:
  - build each index in its own appropriately named directory, e.g.
    - **refs/bowtie2/UCSC/hg19**
    - **refs/bwa/Ensembl/GRCh37**

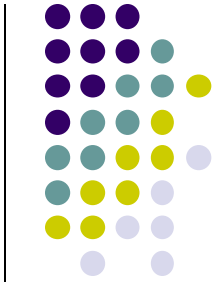
Alignment Workflow



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

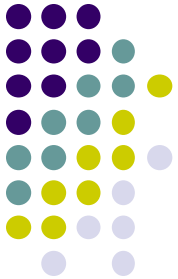
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

SAM file format



- Aligners take FASTQ as input, output alignments in SAM format
  - community file format that describes how reads align to a reference
    - can also include unmapped reads
  - The Bible: http://samtools.github.io/hts-specs/SAMv1.pdf
- SAM file consists of:
  - a header
    - includes reference sequence names and lengths
  - alignment records, one for each sequence read
    - alignments for R1 and R2 reads have separate records, with fields that refer to the mate
    - 11 fixed fields + extensible-format **key:type:value** tuples

SAM file format
Fixed fields



Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{29}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next segment
8	PNEXT	Int	[0,2^{29}-1]	Position of the mate/next segment
9	TLEN	Int	[-2^{29}+1,2^{29}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

read name from fastq

contig + start = locus

insert size, if paired

```
SRR030257.264529      99  NC_012967  1521  29  34M2S  =  1564  79
    CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
    AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4

SRR030257.2669090     147 NC_012967  1521  60  36M    =  1458  -99
    CTGGCCATTATCTCGGTGGTAGGTGATGGTATGCGC
    <<9:<<AAAAAAAAAAAAAAAAAAAAAAAAAAAA
    XT:A:U  NM:i:0  SM:i:37  AM:i:37  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

SAM format – Bitwise flags

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate

1 = part of a read pair
1 = "properly" paired
*1 = read did **not** map*
*1 = mate did **not** map*
1 = minus strand read
1 = mate on minus strand
1 = R1 read
1 = R2 read
1 = secondary possible hit

1 = marked as duplicate

```
                                                                    Decimal      Hex
SRR030257.264529    99  NC_012967  1521  29  34M2S  =  1564  79        99  = 0x63
    CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC                             = 64  = 0x40
    AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,                             + 32  + 0x20
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4  +  2  + 0x02
                                                                     +  1  + 0x01

SRR030257.2669090   147 NC_012967  1521  60  36M    =  1458  -99     147  = 0x93
    CTGGCCATTATCTCGGTGGTAGGTGATGGTATGCGC                           = 128  = 0x80
    <<9:<<AAAAAAAAAAAAAAAAAAAAAAAAAAAA                             +  16  + 0x10
    XT:A:U  NM:i:0  SM:i:37  AM:i:37  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36  +  2  + 0x02
                                                                     +  1  + 0x01
```

http://broadinstitute.github.io/picard/explain-flags.html

Sometimes a CIGAR is just a way of describing how a read is aligned…

```
Ref   CTGGCCATTATCTC--GGTGGTAGGACATGGCATGCCC
Read        aaATGTCGCGGTG.TAGGAggatcc
```

2S5M2I4M1D5M6S

	Op	BAM	Description
	M	0	alignment match (can be a sequence match or mismatch)
	I	1	insertion to the reference
	D	2	deletion from the reference
*	N	3	skipped region from the reference
	S	4	soft clipping (clipped sequences present in SEQ)
*	H	5	hard clipping (clipped sequences NOT present in SEQ)
*	P	6	padding (silent deletion from padded reference)
*	=	7	sequence match
*	X	8	sequence mismatch

"N" indicates splicing event in RNAseq BAMs (e.g. Tophat)

*Rarer / newer

CIGAR = "Concise Idiosyncratic Gapped Alignment Report"

SAM file format
key:type:value tuples

Tag[1]	Type	Description
X?	?	Reserved fields for end users (together with Y? and Z?)
...		
MD	Z	String for mismatching positions. *Regex*: [0-9]+(([A-Z]|\^[A-Z]+)[0-9]+)*[2]
MQ	i	Mapping quality of the mate/next segment
NH	i	Number of reported alignments that contains the query in the current record
NM	i	Edit distance to the reference, including ambiguous bases but excluding clipping
...		

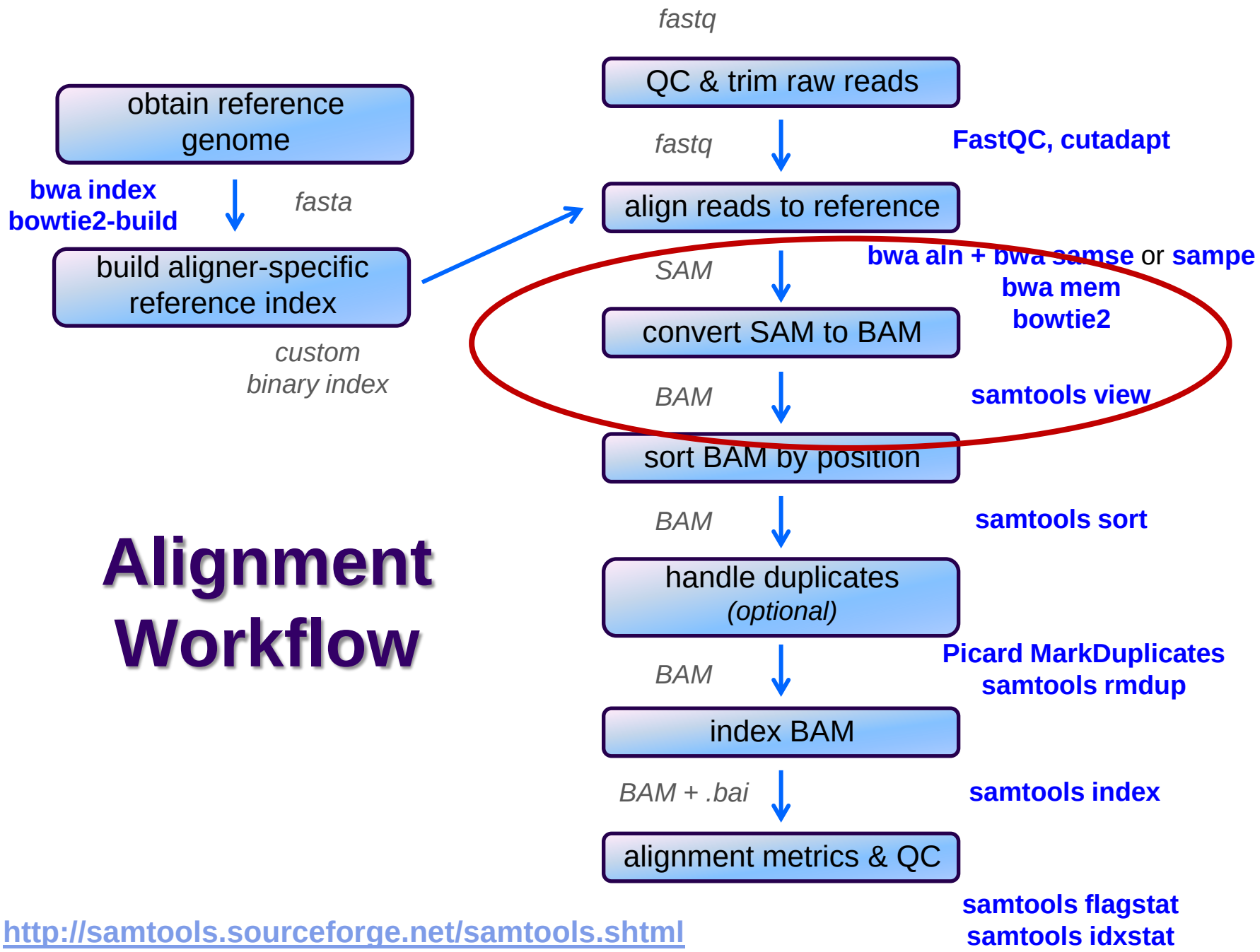
details alignment of query to reference

 # mismatches + insertions + deletions

[2]The MD field aims to achieve SNP/indel calling without looking at the reference. For example, a string '10A5^AC6' means from the leftmost reference base in the alignment, there are 10 matches followed by an A on the reference which is different from the aligned read base; the next 5 reference bases are matches followed by a 2bp deletion from the reference; the deleted sequence is AC; the last 6 bases are matches. The MD field ought to match the CIGAR string.

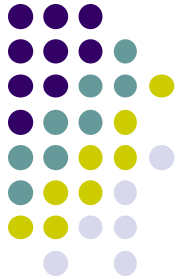
```
SRR030257.264529    99   NC_012967   1521   29   34M2S   =   1564   79
    CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
    AAAAAA:AA;AAAAAA??A%.;?&'3735',()0*,
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4
```

Alignment Workflow

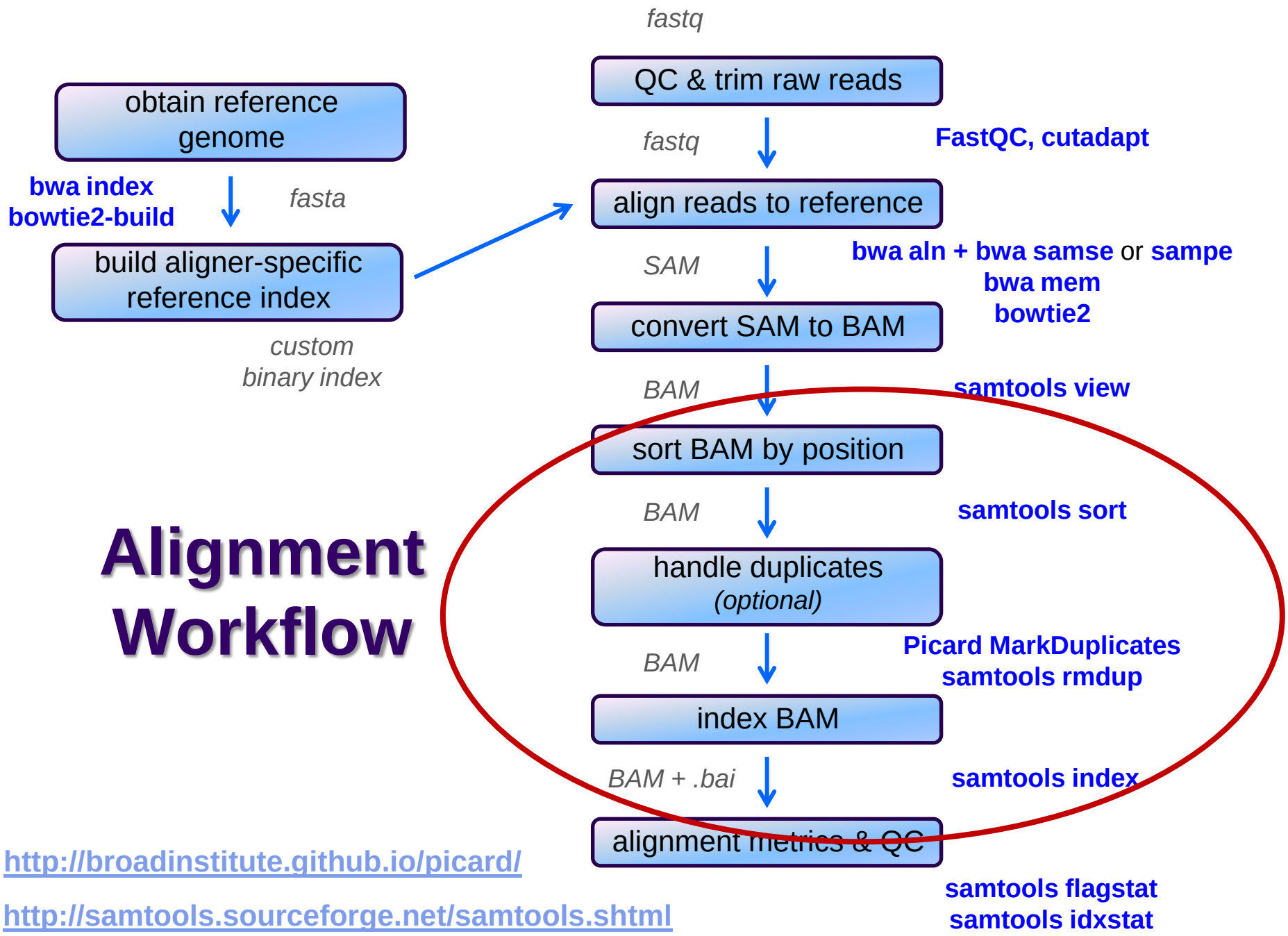


http://samtools.sourceforge.net/samtools.shtml

SAM / BAM files



- SAM and BAM are two forms of the same data
  - SAM – Sequence Alignment Map
    - plain text format
  - BAM – Binary Alignment Map
    - ***same data*** in a custom compressed (**gzip**'d) format
- Differences
  - BAMs are ***much*** smaller than SAM files due to compression
  - BAM files support fast random access; SAM files do not
    - requires the BAM file to be indexed
  - most tools support BAM format and may require indexing
- Best practices
  - remove intermediate SAM and BAM files created during alignment and only save the final sorted, indexed BAM
  - keep your alignment artifacts (BAM, statistics files, log files) separate from the original FASTQ files
    - alignments can be easily re-generated; raw sequences cannot

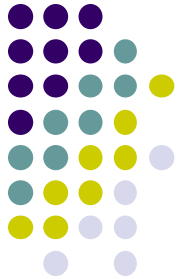


Alignment Workflow

http://broadinstitute.github.io/picard/

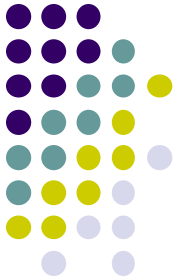
http://samtools.sourceforge.net/samtools.shtml

Sorting / indexing BAM files

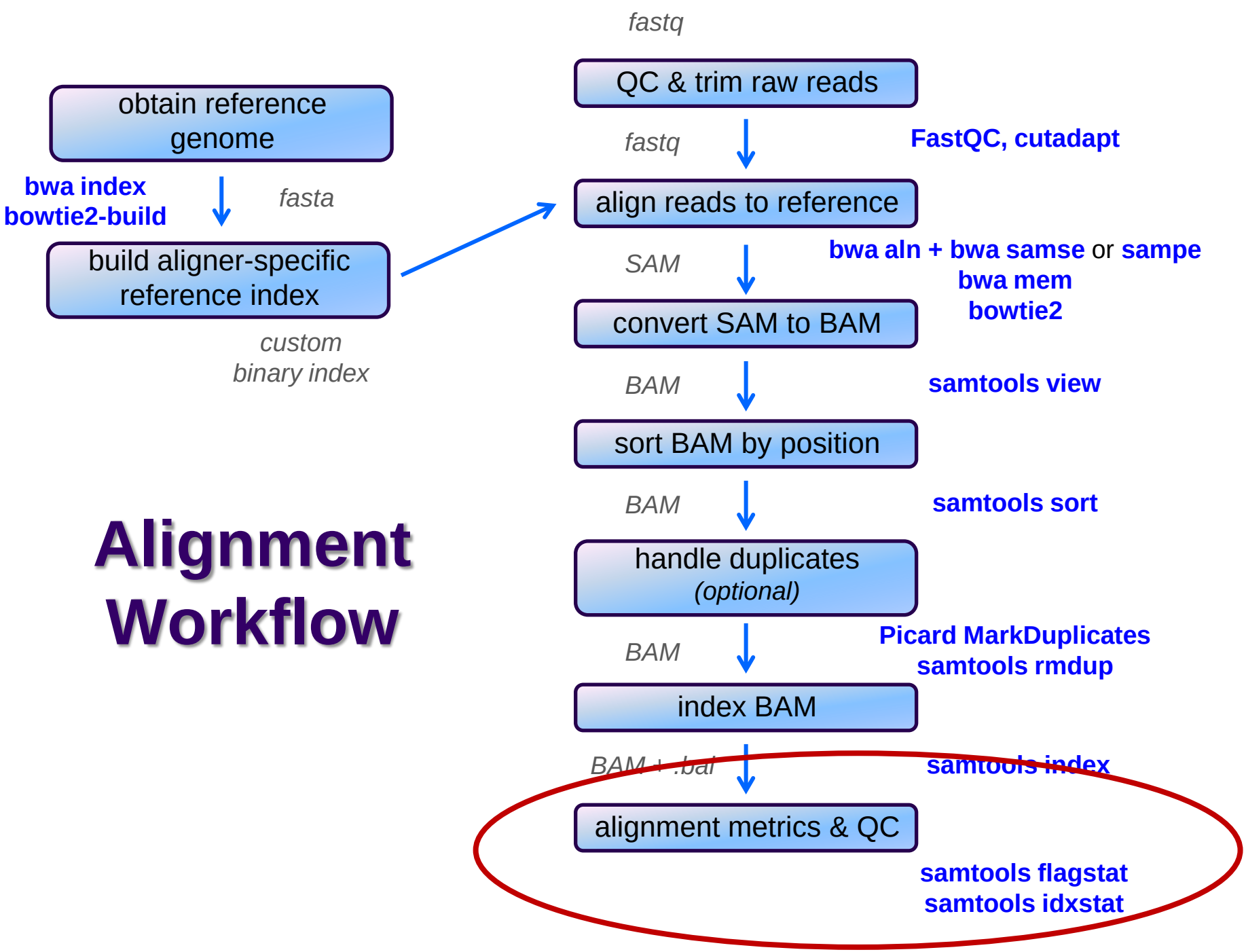


- SAM created by aligner contains read records in ***name order***
  - same order as read names in the input FASTQ file
    R1, R2 have adjacent SAM records
  - SAM → BAM conversion does not change the name-sorted order
- Sorting BAM puts records in ***locus order***
  - by contig name then start position
    - contig name order given in SAM/BAM header
    - based on order of sequences in FASTA used to build reference
  - sorting is *very* compute and I/O intensive
    - can take several hours for large BAM
- Indexing a locus-sorted BAM allows fast random access
  - creates a binary alignment index file **(.bai)**
  - quite fast

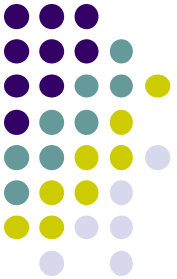
Handling Duplicates



- Optional step, but very important for many protocols
- Definition of duplicate alignments:
  - single end reads or singleton/discordant alignment
    - alignments start at the same location and have the same length
  - properly paired reads
    - pairs have same external coordinates
- Two choices for handling:
  - **samtools rmdup** – ***removes*** duplicates entirely
    - faster, but data is lost
    - does not properly handle data from multiple lanes
  - **Picard MarkDuplicates** – ***flags*** duplicates only
    - slower, but all alignments are retained
    - alignments from different lanes/replicates are handled properly
  - both tools are quirky in their own ways


fastq
obtain reference genome
QC & trim raw reads
fastq
FastQC, cutadapt
bwa index
bowtie2-build
fasta
align reads to reference
build aligner-specific reference index
SAM
bwa aln + bwa samse or sampe
bwa mem
bowtie2
convert SAM to BAM
custom binary index
BAM
samtools view
sort BAM by position
BAM
samtools sort
handle duplicates (optional)
BAM
Picard MarkDuplicates
samtools rmdup
index BAM
BAM + .bai
samtools index
alignment metrics & QC
samtools flagstat
samtools idxstat
Alignment Workflow

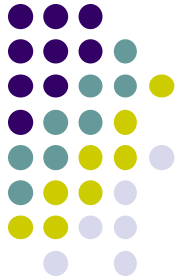
Alignment metrics



- **samtools flagstat**
  - simple statistics based on alignment record flag values
    - total sequences (R1+R2), total mapped
    - number properly paired
    - number of duplicates (0 if duplicates were not marked)

```
30146426 + 0 in total (QC-passed reads + QC-failed reads)
13532165 + 0 duplicates
28804693 + 0 mapped (95.55%:-nan%)
30146426 + 0 paired in sequencing
15073213 + 0 read1
15073213 + 0 read2
28546786 + 0 properly paired (94.69%:-nan%)
28712992 + 0 with itself and mate mapped
91701 + 0 singletons (0.30%:-nan%)
64973 + 0 with mate mapped to a different chr
50382 + 0 with mate mapped to a different chr (mapQ>=5)
```

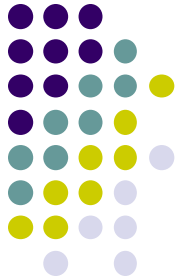
Computing average insert size



- Needed for RNAseq alignment using **tophat**
- Simple **awk** script that computes average insert size for a BAM
  - **-F 0x4** filter to **samtools view** says only consider mapped reads
    - technically “not unmapped”
  - the **-f 0x2** filter says consider only properly paired reads
    - they have reliable “insert size" values in column 9
  - insert size values are negative for minus strand reads
    - can ignore because each proper pair should have one plus and one minus strand alignment

```
samtools view -F 0x4 -f 0x2 my_pe_data.bam | awk \
  'BEGIN{ FS="\t"; sum=0; nrec=0; }
   { if ($9 > 0) {sum += $9; nrec++;} }
   END{ print sum/nrec; }'
```

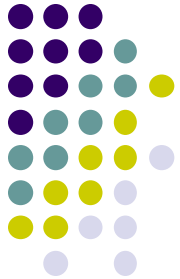
Interpreting alignment metrics



- Table below is taken from a spreadsheet I keep on all our alignments
  - all are yeast paired-end read datasets from ChIP-seq experiments
- Alignment rates
  - samples 1-3 have excellent alignment rates & good rates of proper pairing
  - sample 4
    - has an unusually low alignment rate for a ChIP-seq dataset
    - has a median insert size of only 109, and these were un-trimmed 50 bp reads
    - could 3' adapter contamination be affecting the alignment rate?
      - try re-aligning the sequences after trimming, say to 30 bases
      - see if the alignment rate improves

#	totSeq	totAlign	% align	numPair	pePrAln	% prPr	nDup	% dup	multiHit	% multi	iszMed
1	149,644,822	145,228,810	97.0%	74,822,411	72,221,545	96.5%	49,745,225	34%	16,216,807	11%	181
2	981,186	860,940	87.7%	490,593	424,915	86.6%	609,378	71%	127,987	15%	148
3	22,573,348	21,928,789	97.1%	11,286,674	10,783,971	95.5%	9,408,725	43%	3,711,004	17%	132
4	7,200,628	3,460,992	48.1%	3,600,314	1,626,121	45.2%	1,234,524	36%	649,690	19%	109

Interpreting alignment metrics

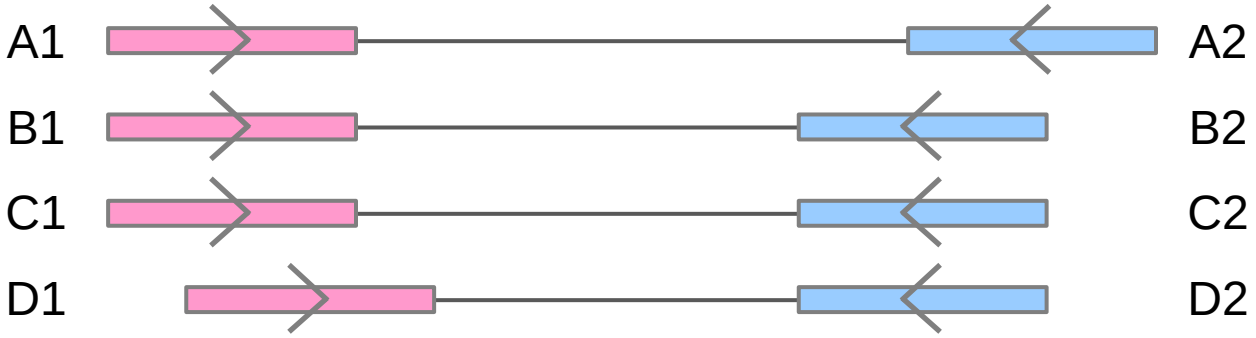


- Duplication rates
  - sample 1 is incredibly deeply sequenced (yeast genome only ~12 Mbase)
    - has a *very* low duplication rate considering
    - turns out this is a control dataset (Mock ChIP), so is a great control to use (wonderfully complex!)
  - sample 2 is not very deeply sequenced but has a high duplication rate (71%)
    - subtracting duplicates from total aligned leaves only ~250,000 non-dup reads
      - not enough for further analysis (prefer 500,000+)
  - sample 3 has reasonable sequencing depth with substantial duplication (43%)
    - still leaves plenty of non-duplicate reads (> 12 million)

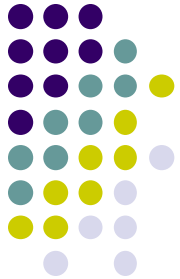
#	totSeq	totAlign	% align	numPair	pePrAln	% prPr	nDup	% dup	multiHit	% multi	iszMed
1	149,644,822	145,228,810	97.0%	74,822,411	72,221,545	96.5%	49,745,225	34%	16,216,807	11%	181
2	981,186	860,940	87.7%	490,593	424,915	86.6%	609,378	71%	127,987	15%	148
3	22,573,348	21,928,789	97.1%	11,286,674	10,783,971	95.5%	9,408,725	43%	3,711,004	17%	132
4	7,200,628	3,460,992	48.1%	3,600,314	1,626,121	45.2%	1,234,524	36%	649,690	19%	109

Read vs fragment duplication

- Consider the 4 fragments below
  - 4 R1 reads (pink), 4 R2 reads (blue)
- Duplication when only 1 end considered
  - A1, B1, C1 have identical sequences, D1 different
    - 2 unique + 2 duplicates = 50% duplication rate
  - B2, C2, D2 have identical sequences, A2 different
    - 2 unique + 2 duplicates = 50% duplication rate
- Duplication when both ends considered
  - fragments B and C are duplicates (same external sequences)
    - 3 unique + 1 duplicate = 25% duplication rate



Alignment wrap up



- Many tools involved
  - choose one (or two) and learn their options well
- Many steps are involved in the full alignment workflow
  - important to go through manually a few times for learning
    - but gets tedious quickly!
  - best practice
    - automate series of complex steps by wrapping into a ***pipeline script***
    - e.g. **bash**, **perl** or **python** script
- For UT folks with TACC accounts
  - I have a set of TACC-aware alignment pipeline scripts
    - plus a set of pre-build reference indexes

Final thoughts

- Good judgement comes from experience

  unfortunately…
- Experience comes from bad judgement!
- So go get started making your 1st 1,000 mistakes….