Preprocessing (from reads to a count matrix)

Roberta Menafra
14-10-2019

Bioinformatician LGTC (Leiden Genome Technology Center)

Leids Universitair Medisch Centrum

Preprocessing (from reads to a count matrix)

- Sequencing data format
- Data pre-processing
- 10X pipeline (Cell Ranger)
  - mkfastq
  - count
- Results examples

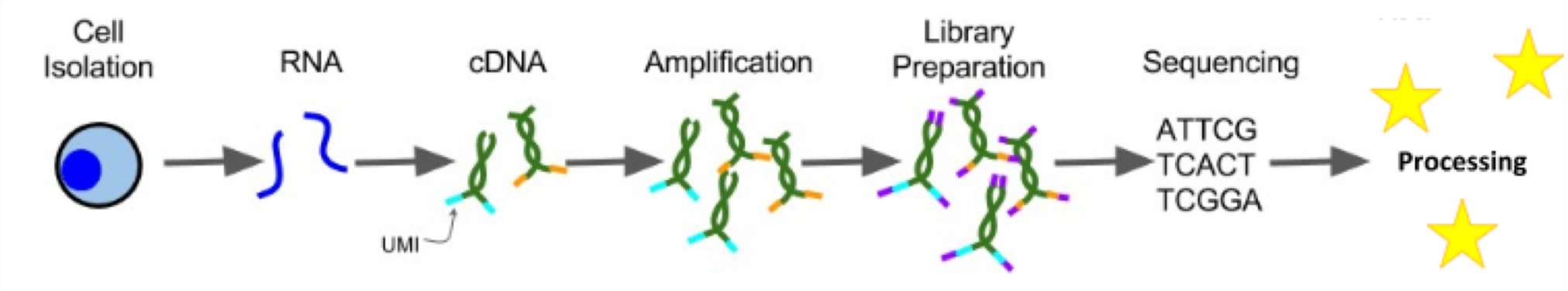
Common file formats in NGS

- **bcl**
- **fastq**
- **bam**
- mtx, tsv
- hdf5 (.h5, .h5ad)

BCL

Raw data files in binary base call format. This sequencing file format requires conversion to FASTQ format for use with user-developed or third-party data analysis tools.
Illumina offers bcl2fastq Conversion Software to convert BCL files. bcl2fastq is included, standalone conversion software that demultiplexes data and converts BCL files to standard FASTQ file formats for downstream analysis.

From sample to raw data



Data processing steps

1. Create Fastq files
2. Fastq quality control
3. Align fastq

Samples demultiplexing

Sequence runs on NGS instruments are typically carried out with multiple samples pooled together. An **index tag** (also called a barcode) consisting of a unique sequence (between 6 and 12bp) is added to each sample so that the sequence reads from different samples can be identified.

The other requirement is a sample sheet – a simple comma separated file (csv) with the library chemistry, sample names and the index tag used for each sample

```
[Header],,,,,,,,,
IEMFileVersion,4,,,,,,,
Date,20-10-2014,,,,,,,
Workflow,GenerateFASTQ,,,,,,,
Application,FASTQ Only,,,,,,,
Assay,Nextera,,,,,,,
Description,,,,,,,,
Chemistry,Amplicon,,,,,,,
,,,,,,,,,
[Reads],,,,,,,,,
151,,,,,,,,,
151,,,,,,,,,
,,,,,,,,,
[Settings],,,,,,,,,
ReverseComplement,0,,,,,,,
Adapter,,,,,,,,,
,,,,,,,,,
[Data],,,,,,,
Lane,Sample_ID,Sample_Name,Sample_Plate,Sample_Well,I7_Index_ID,index,I5_Index_ID,index2
4,AV_1_HTO,AV_1_HTO,,,,ATTACTCG,,,
5,AV_1_HTO,AV_1_HTO,,,,ATTACTCG,,,
```

```
bcl2fastq --runfolder-dir 190826_E00603_0316_AH3GW3CCX2/ --output-dir HTO/ --sample-sheet samples_HTO.csv --barcode-mismatches 0
```

```
output-dir: HTO/Reports/html/flowcellID/all/all/all/laneBarcode.html
```

Flowcell Summary

Clusters (Raw)	Clusters(PF)	Yield (MBases)
1,242,422,208	946,733,762	285,914

Lane Summary

Lane	Project	Sample	Barcode sequence	PF Clusters	% of the lane	% Perfect barcode	% One mismatch barcode	Yield (Mbases)	% PF Clusters	% >= Q30 bases	Mean Quality Score
4	default	AV_1_HTO	ATTACTCG	54,321,382	11.52	100.00	NaN	16,405	100.00	46.78	26.39
4	default	Undetermined	*unknown*	417,222,614	88.48	100.00	NaN	126,001	73.60	67.87	31.88
5	default	AV_1_HTO	ATTACTCG	54,933,100	11.56	100.00	NaN	16,590	100.00	46.78	26.40
5	default	Undetermined	*unknown*	420,256,666	88.44	100.00	NaN	126,918	74.21	67.77	31.85

Top Unknown Barcodes

Lane	Count	Sequence	Lane	Count	Sequence
4	116,377,900	AGTGGAAC	5	116,758,060	AGTGGAAC
	107,277,740	GTCTCCTT		107,610,520	GTCTCCTT
	79,994,540	TCACATCA		80,406,280	TCACATCA
	74,171,580	CAGATGGG		74,495,820	CAGATGGG
	19,713,380	CAAAAGAT		19,888,220	CAAAAGAT
	705,960	GTCTCCTA		706,360	GTCTCCTA
	629,960	TCACTCAA		638,440	TCACTCAA
	600,500	CAGATGGA		606,040	AGTGAACA
	597,720	AGTGAACA		603,420	CAGATGGA
	564,060	TCCATCAA		574,720	TCCATCAA

FASTQ

- NGS data is often in FASTQ format
- FASTQ is a text-based format for storing both sequence and its corresponding quality score
- Four lines per sequence (read)
  - @ followed by the unique sequence identifier
  - The nucleotide sequence
  - + The quality line break
  - The quality scores in ASCII characters

View FASTQ Files

Viewing entire file

cat file1.fastq

Viewing first 10 lines

head file1.fastq

```
@A00379:133:HMWGLDSXX:1:1101:1163:1000 2:N:0:GAGGATCT
AAAAAAAAATAAAAAAAAATAAAAAAAATGATAAAAAAAATAAAATTAAAAAATATAAAAAAAAAAAAATTTTTTTTTTATTAAGTAAAAAAAATTAAAAATAAATTAAAAAAAAAAAAAAA
+
FF,:::FF,,F,:F,FF,F:FFFF:F:F,,F:F:FFFFF,,:F:,,,:FF:F,,,,F,FFF:FFFF,,,,:,F,,,,:,,,:F,,F::F,F,,,,F::F,FFF,,,,:,FFFFFF,FFFF:
@A00379:133:HMWGLDSXX:1:1101:1253:1000 2:N:0:GAGGATCT
GATGGTGTAATGCCTGTCTTTAAGTACAAACTGCAAAGTGAAGCCACCCAGATTTATTTCCTTTCTCTGGACCAAGTGTCCCAAACTGAACACTGAGACTGAGGCAGATTTGAAATGATCCA
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFF:FFFFFFFF:FFFFFFFFFFFFFFFFFFFFFFFFFFFFF,,FFFFFFFFFFFFFFFF:FFFFF,FF:FFFFFFFFFF,FF,FFFF
```

FASTQ

```
@A00379:133:HMWGLDSXX:1:1101:1163:1000 2:N:0:GAGGATCT
+
FF,:::FF,,F,:F,FF,F:FFFF:F:F,,F:F:FFFFF,,:F:,,,:FF:F,,,,F,FFF:FFFF,,,,:,F,,,:,,,:F,,F::F,F,,,F::F,FFF,,,,:,FFFFFF,FFFF::,:
```

Instrument — RunID — FlowcellID — Read Number (Paired 2/2) — Index Sequence

Table 1 ASCII Characters Encoding Q-scores 0–40

Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score
!	33	0	/	47	14	=	61	28
"	34	1	0	48	15	>	62	29
#	35	2	1	49	16	?	63	30
$	36	3	2	50	17	@	64	31
%	37	4	3	51	18	A	65	32
&	38	5	4	52	19	B	66	33
'	39	6	5	53	20	C	67	34
(	40	7	6	54	21	D	68	35
)	41	8	7	55	22	E	69	36
*	42	9	8	56	23	F	70	37
+	43	10	9	57	24	G	71	38
,	44	11	:	58	25	H	72	39
-	45	12	;	59	26	I	73	40
.	46	13	<	60	27			

The quality score is associated to a probability of error, of an incorrect base call

The sequencing quality score of a given base, Q, is defined by the following equation:

$Q = -10\log_{10}(e)$

where **e** is the **estimated probability** of the base call being wrong.

- **Higher Q scores** indicate a smaller probability of error.
- **Lower Q scores** can result in a significant portion of the reads being unusable.
- Quality scores are generally in the range of 0-40

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

FASTQC

Before analysing the data to draw biological conclusions you should always perform some simple quality control checks to ensure that the raw data looks good and there are no problems or biases in your data which may affect how you can usefully use it.

FastQC aims to provide a QC report which can spot problems which originate either in the sequencer or in the starting library material

FastQC Report

Read1

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	Pool_1_S1_L001_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	380844038
Sequences flagged as poor quality	0
Sequence length	26
%GC	51

FastQC Report

Read2

Summary

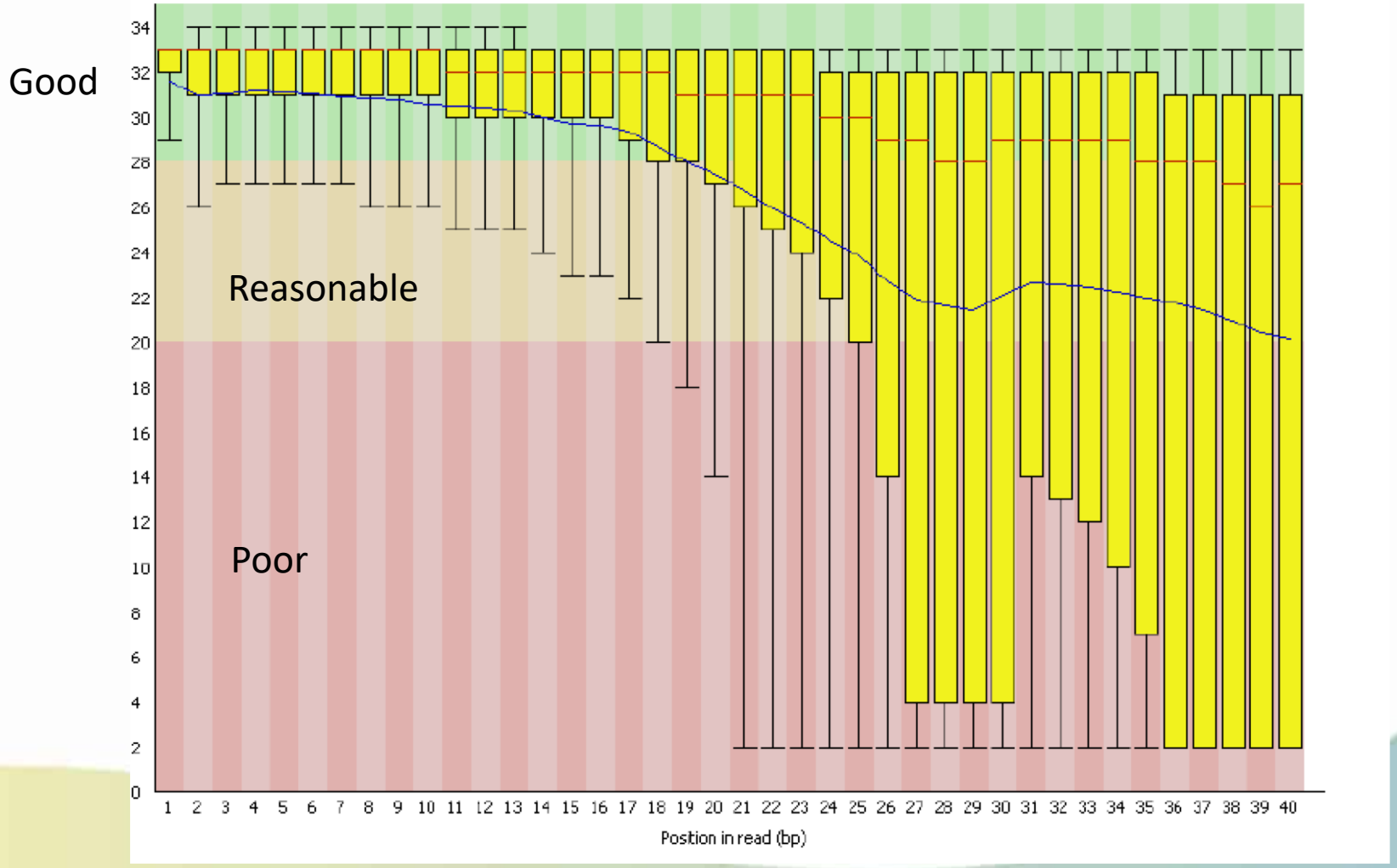
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	Pool_1_S1_L001_R2_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	380844038
Sequences flagged as poor quality	0
Sequence length	150
%GC	36

Per Base Sequence Quality

This view shows an overview of the range of quality values across all bases at each position in the FastQ file.



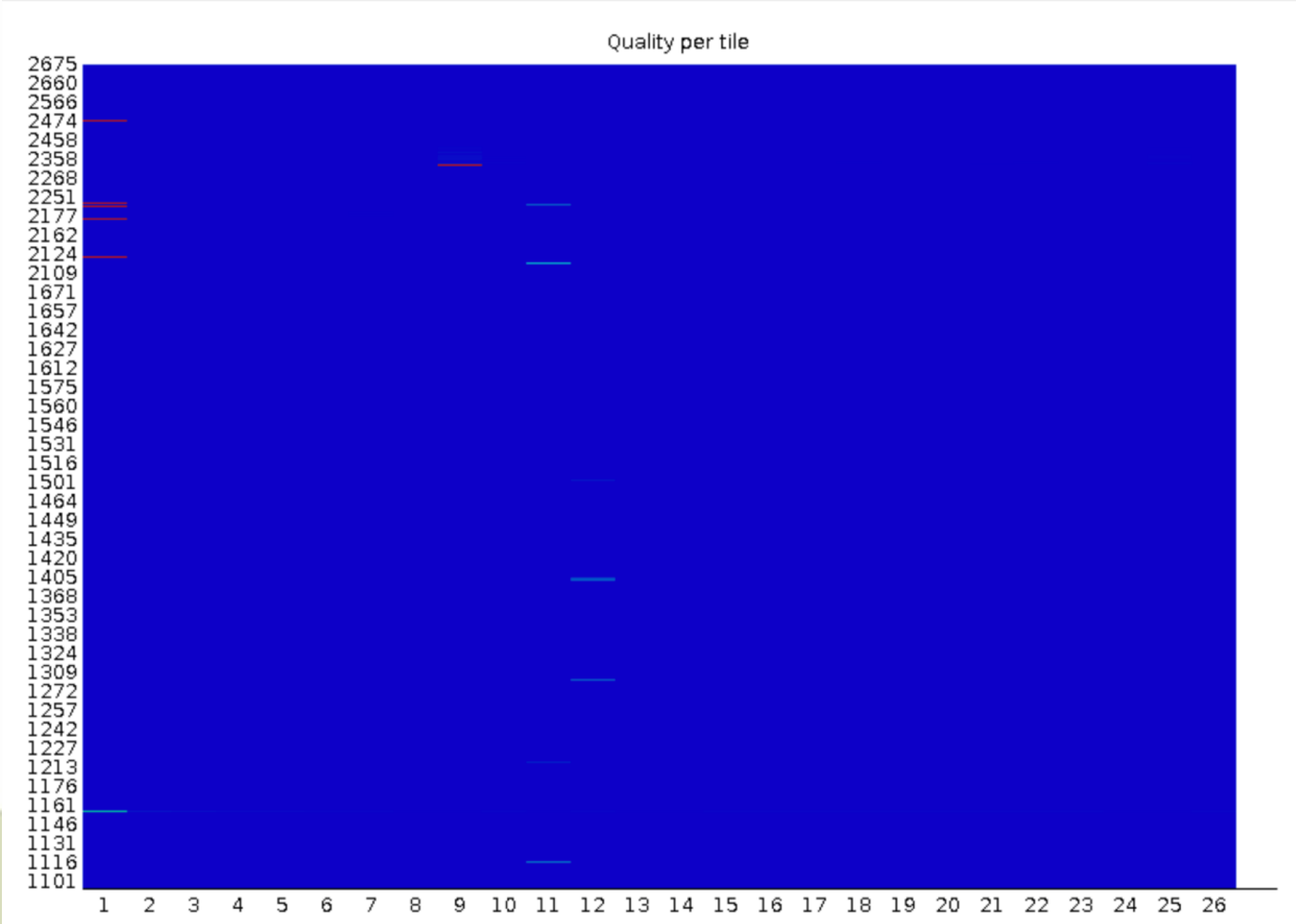
Per Tile Sequence Quality

The graph allows you to look at the quality scores from each tile across all of your bases to see if there was a loss in quality associated with only one part of the flowcell.

The plot shows the deviation from the average quality for each tile. The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse qualities than other tiles for that base. A good plot should be blue all over.

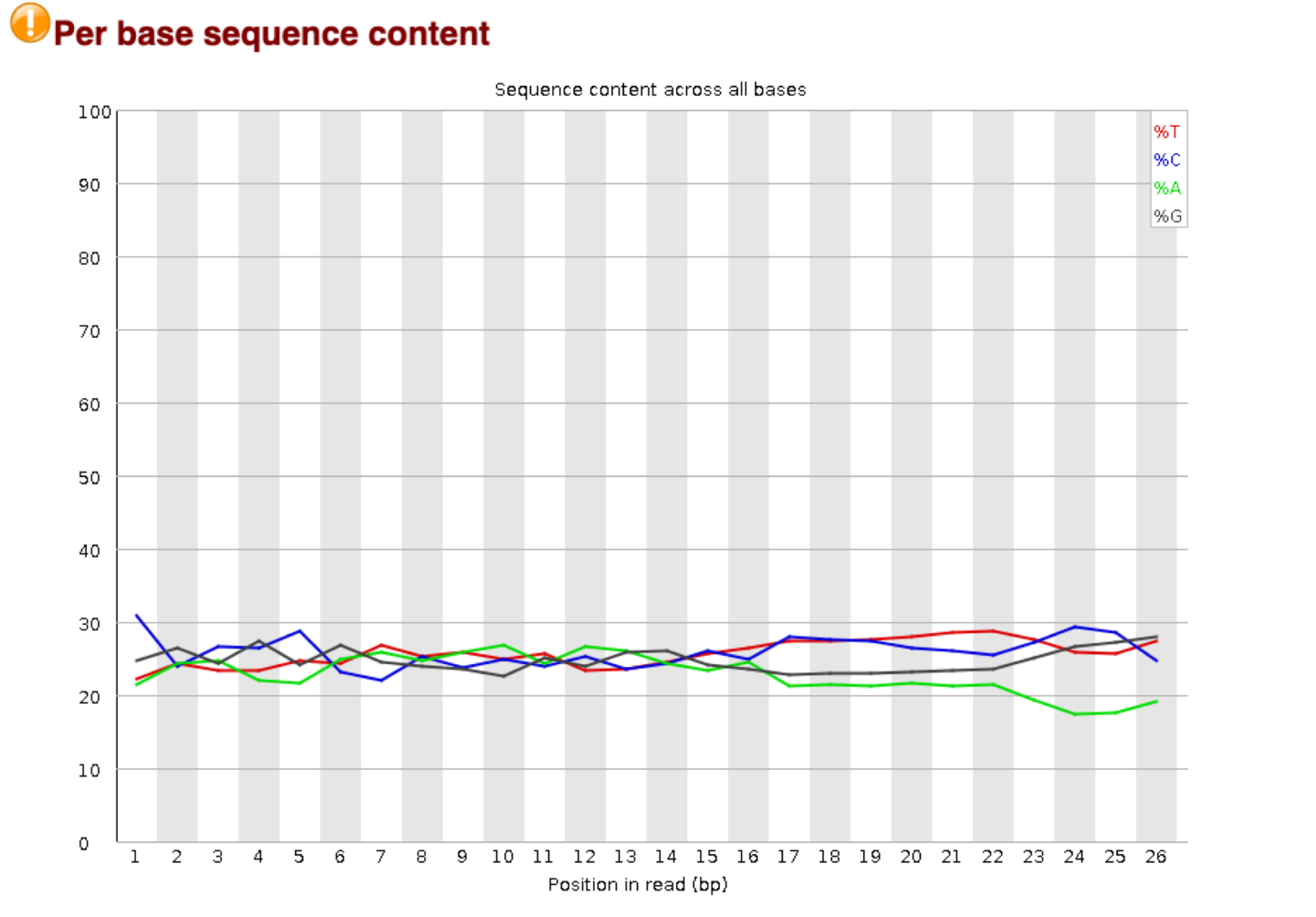
Common reasons for warnings

It has been observed that greater variation in the phred scores attributed to tiles can also appear when a flowcell is generally overloaded. In this case events appear all over the flowcell rather than being confined to a specific area or range of cycles. We would generally ignore errors which mildly affected a small number of tiles for only 1 or 2 cycles.



Per Sequence Quality Scores

The per sequence quality score report allows you to see if a subset of your sequences have universally low quality values. It is often the case that a subset of sequences will have universally poor quality, often because they are poorly imaged (on the edge of the field of view etc), however these should represent only a small percentage of the total sequences.



Warning: This module issues a warning if the difference between A and T, or G and C is greater than 10% in any position.
Failure: This module will fail if the difference between A and T, or G and C is greater than 20% in any position.

Overrepresented Sequences

A normal high-throughput library will contain a diverse set of sequences, with no individual sequence making up a tiny fraction of the whole. Finding that a single sequence is very overrepresented in the set either means that it is highly biologically significant, or indicates that the library is contaminated, or not as diverse as you expected.
This module lists all of the sequence which make up more than 0.1% of the total. Hits must be at least 20bp in length and have no more than 1 mismatch. Finding a hit doesn't necessarily mean that this is the source of the contamination, but may point you in the right direction. It's also worth pointing out that many adapter sequences are very similar to each other so you may get a hit reported which isn't technically correct, but which has very similar sequence to the actual match.

Warning

This module will issue a warning if any sequence is found to represent more than 0.1% of the total.

Failure

This module will issue an error if any sequence is found to represent more than 1% of the total.

Typical artifacts

Primers, sequence adapters

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTT	103227946	27.105044506433888	No Hit
GCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTT	8592717	2.256229884843307	No Hit
AGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTT	4198600	1.1024460359282295	No Hit
GTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTT	1116215	0.2930897923102055	No Hit
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	1018762	0.26750110238039226	No Hit
CAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTT	1005203	0.26394085234439196	No Hit
AGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTT	931329	0.24454341070714097	No Hit
GGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	563221	0.14788757176238113	No Hit
AAGCAGTGGTATCAACGCAGAGTATTTTTTTTTTTTTTTTTTTTTTTTT	392982	0.1031871214431352	No Hit

Sequence filtering

Adapter trimming
Quality trimming
Trimming fixed length

Tools

Cutadapt
SeqTK
Trimmomatic

10X technology and Cell Ranger

Cell Ranger is a set of analysis pipelines that process Chromium single-cell RNA-seq output to align reads, generate gene-barcode matrices and perform clustering and gene expression analysis. Cell Ranger includes four pipelines relevant to single-cell gene expression experiments:

•**cellranger mkfastq** demultiplexes raw base call (BCL) files generated by Illumina sequencers into FASTQ files. It is a wrapper around Illumina's bcl2fastq, with additional useful features that are specific to 10x libraries and a simplified sample sheet format.

•**cellranger count** takes FASTQ files from cellranger mkfastq and performs alignment, filtering, barcode counting, and UMI counting. It uses the Chromium cellular barcodes to generate feature-barcode matrices, determine clusters, and perform gene expression analysis. The count pipeline can take input from multiple sequencing runs.

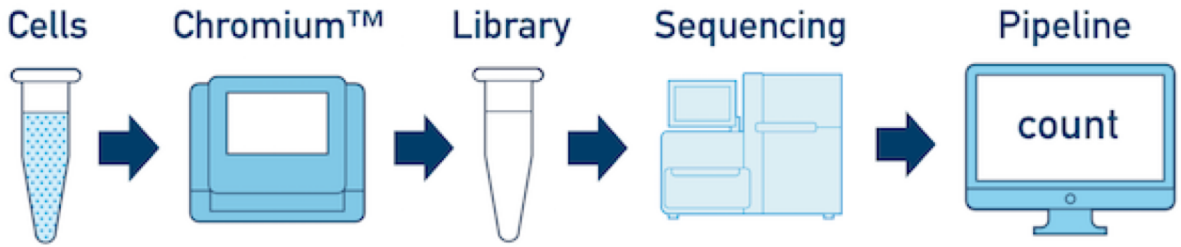
•**cellranger aggr** .
•**cellranger reanalyze**

Cell Ranger mkfastq

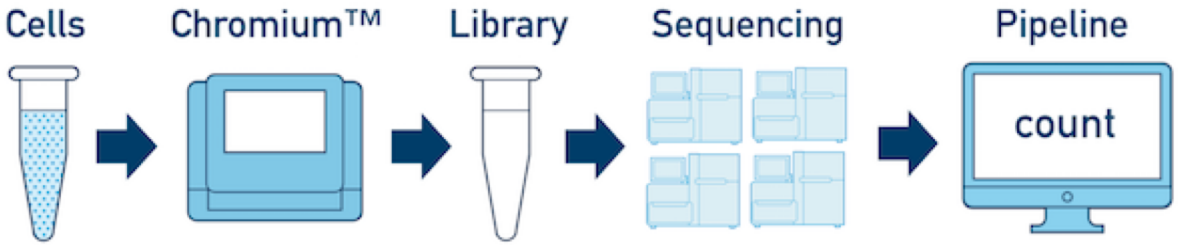
10X Data demultiplexing --> cellranger mkfastq

Workflows

The Cell Ranger workflow always starts with running `cellranger mkfastq` on each flowcell. The subsequent steps vary depending on how many samples, libraries and flowcells you have. We will describe them in order of increasing complexity:



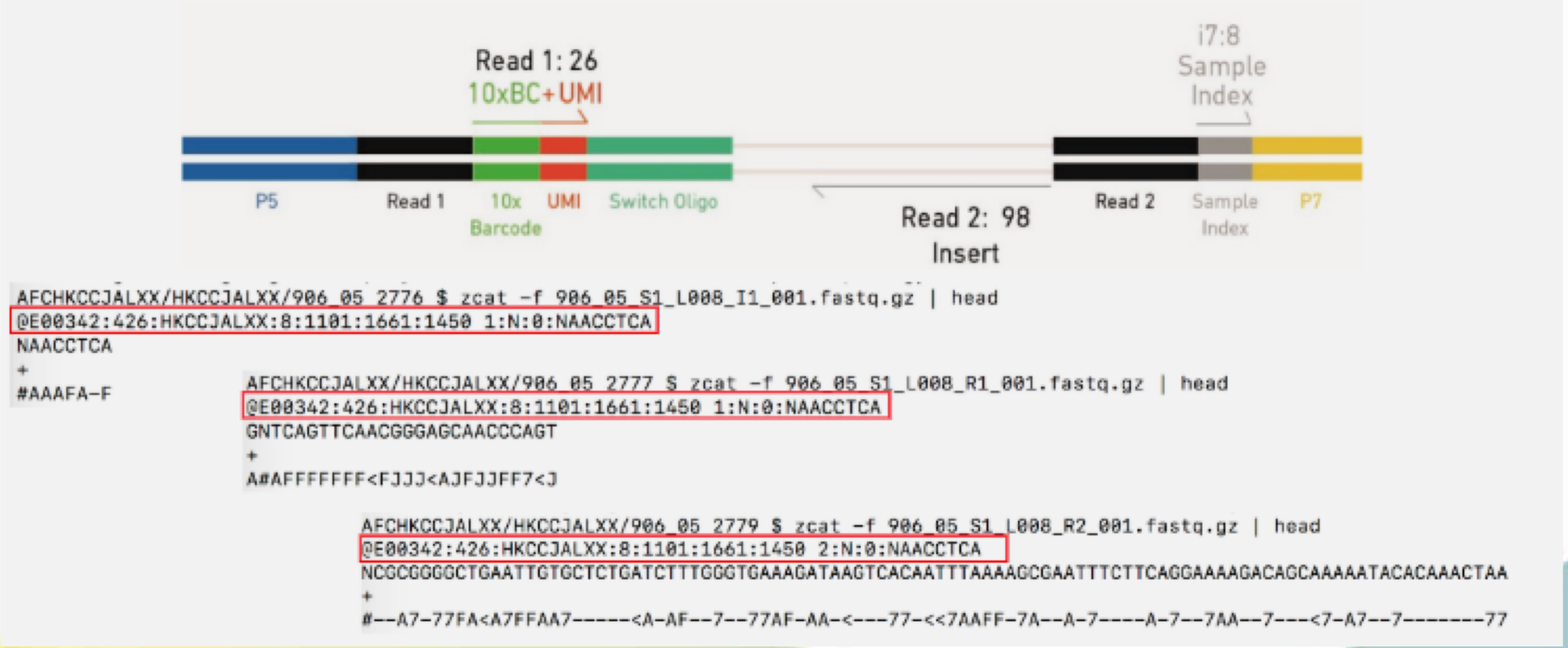
Single Sample, Library, and Flowcell is the most basic case. You have a single biological sample, which was prepared into a single library, and then sequenced on a single flowcell. Assuming the FASTQs have been generated with cellranger mkfastq, you just need to run cellranger count as described in Single-Library Analysis.



One Library, Multiple Flowcells If you have a library which was sequenced across multiple flowcells, you can pool the reads from both sequencing runs. Follow the steps in Multi-Flowcell Samples to combine them in a single cellranger count run.

Cell Ranger mkfastq

Cellranger mkfastq produces :
R1.fastq
R2.fastq
Index.fastq



Data processing steps

- ✓ Create Fastq files
- ✓ Fastq quality control
3. Align fastq

Aligning reads to a reference

FASTQ files contain sequence information that we wish to map to genes in a genome:
1.Select your genome
2.Select your gene annotation file (generally gtf format)
3.Run the alignment program
4.Result of alignment is generally stored in a sam/bam file

SAM Format

This is the most basic, human readable format, generated by almost every alignment algorithm that exists. It consists of a header, a row for every read in your dataset, and 11 tab-delimited fields describing that read.

SAM Header

The header varies in size but adheres to a particular format depending on what information you decide to add. Some example information that can be entered into the header is: command that generated the SAM file, SAM format version, sequencer name and version. The full list of available header fields can be found below

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,254}`	Query template NAME
2	FLAG	Int	$[0, 2^{16}-1]$	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	$[0, 2^{31}-1]$	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 read
8	PNEXT	Int	$[0, 2^{31}-1]$	Position of the mate/next read
9	TLEN	Int	$[-2^{31}+1, 2^{31}-1]$	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

http://samtools.github.io/hts-specs/SAMv1.pdf

Bitwise Flag

The bitwise flag is a lookup code to explain certain features about the particular read (exact same concept as Linux permission codes!). It tells you whether the read aligned, is marked a PCR duplicate, if it's mate aligned, etc. and any combination of the available tags, seen below:

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

One important thing to note is that any combination of these flags results in one integer, which makes interpreting it a bit difficult. To make it easy you can check to either encode or decode a bitwise flag.
https://broadinstitute.github.io/picard/explain-flags.html

Decoding SAM flags

This utility makes it easy to identify what are the properties of a read based on its SAM flag value, or conversely, to find what the SAM Flag value would be for a given combination of properties.

To decode a given SAM flag value, just enter the number in the field below. The encoded properties will be listed under Summary below, to the right.

SAM Flag: 4 Explain

Switch to mate Toggle first in pair / second in pair

Find SAM flag by property:

To find out what the SAM flag value would be for a given combination of properties, tick the boxes for those that you'd like to include. The flag value will be shown in the SAM Flag field above.

- [ ] read paired
- [ ] read mapped in proper pair
- [x] read unmapped
- [ ] mate unmapped
- [ ] read reverse strand
- [ ] mate reverse strand
- [ ] first in pair
- [ ] second in pair
- [ ] not primary alignment
- [ ] read fails platform/vendor quality checks
- [ ] read is PCR or optical duplicate
- [ ] supplementary alignment

Summary:

read unmapped (0x4)

MapQ (Mapping Quality)

This value reports how well the read aligned to the reference. Different algorithms report it differently but nonetheless, the greater the number the better the alignment (generally).

CIGAR String

This is a shorthand way to encode an entire alignment. Instead of writing the whole alignment out, operators have been defined and are used in combination with numbers to explain which part of the sequence aligns, which doesn't, and everything in between. The definition for the operators can be found here:

Op	Description
M	Match (alignment column containing two letters). This could contain two different letters (mismatch) or two identical letters. USEARCH generates CIGAR strings containing Ms rather than X's and ='s (see below).
D	Deletion (gap in the target sequence).
I	Insertion (gap in the query sequence).
S	Segment of the query sequence that does not appear in the alignment. This is used with soft clipping, where the full-length query sequence is given (field 10 in the SAM record). In this case, S operations specify segments at the start and/or end of the query that do not appear in a local alignment.
H	Segment of the query sequence that does not appear in the alignment. This is used with hard clipping, where only the aligned segment of the query sequences is given (field 10 in the SAM record). In this case, H operations specify segments at the start and/or end of the query that do not appear in the SAM record.
=	Alignment column containing two identical letters. USEARCH can read CIGAR strings using this operation, but does not generate them.
X	Alignment column containing a mismatch, i.e. two different letters. USEARCH can read CIGAR strings using this operation, but does not generate them.

samtools view aligned_reads.sam | head –n 1

HS2000-940_146:5:1101:1161:63226 73 NC_000020.11 23775298 60 78M22S = 23775298 0 CTGNTAGCCCTGCTGAATCTCCCTCCTGACCCAACTCCCTCNTNNNNNNNGCTGGGTGACTGCTGNCNNCACNGGCTGTGNNNNNNNNNNNNNNCAGCTG G ?@@#4ADDDFDFFHIGGFCFHCHFGIHGCGHEHHEHD3?BH#0#######--5CEECG=?AEEHE################################## NM:i:13 MD:Z:3G37C1C0T0A0C0T0C0T15T1C0T3T5 AS:i:52 XS:i:0 RG:Z:sample_1 HS2000-940_146:5:1101:1161:63226 133 NC_000020.11 23775298 0 * = 23775298 0 NNCTCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGNNNNCNAA AGGAGCCTGGGT ## AS:i:0 XS:i:0 RG:Z:sample_1 HS2000-940_146:5:1101:1262:12434 99 NC_000020.11 23843774 60 100M = 23843977 258

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,254}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{31}-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 read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

BAM format

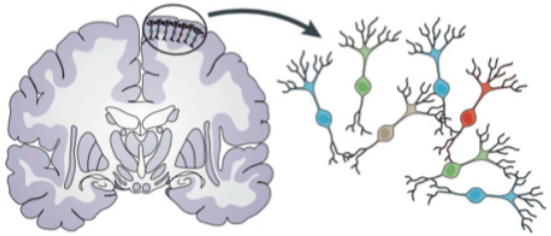
This is the same format except that it encoded in binary which means that it is significantly smaller than the SAM files and significantly faster to read, though it is not human legible and needs to be converted to another format (i.e. SAM) in order to make sense to us.
Some special tools are needed in order to make sense of BAM, such as Samtools, Picard Tools,

View BAM Files

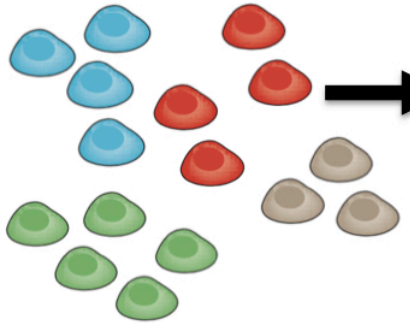
```
samtools view alignment.bam | head
```

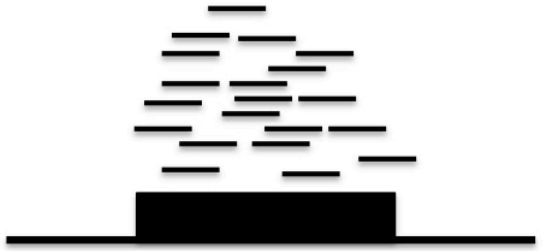
Single-cell RNA-Seq (scRNA-Seq)

Tissue (e.g. tumor)



Isolate and sequence individual cells



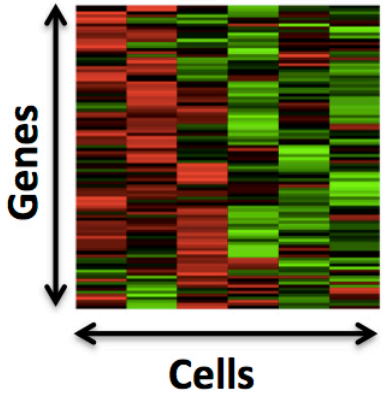


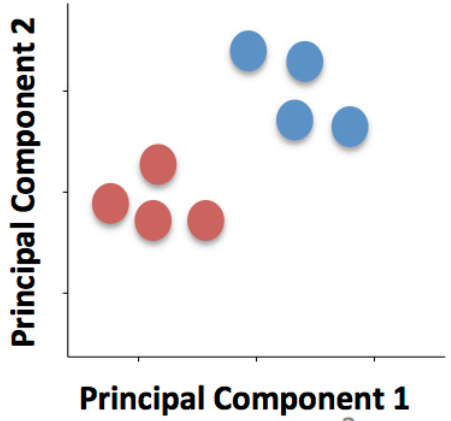
Gene 1
Cell 1

Read Counts

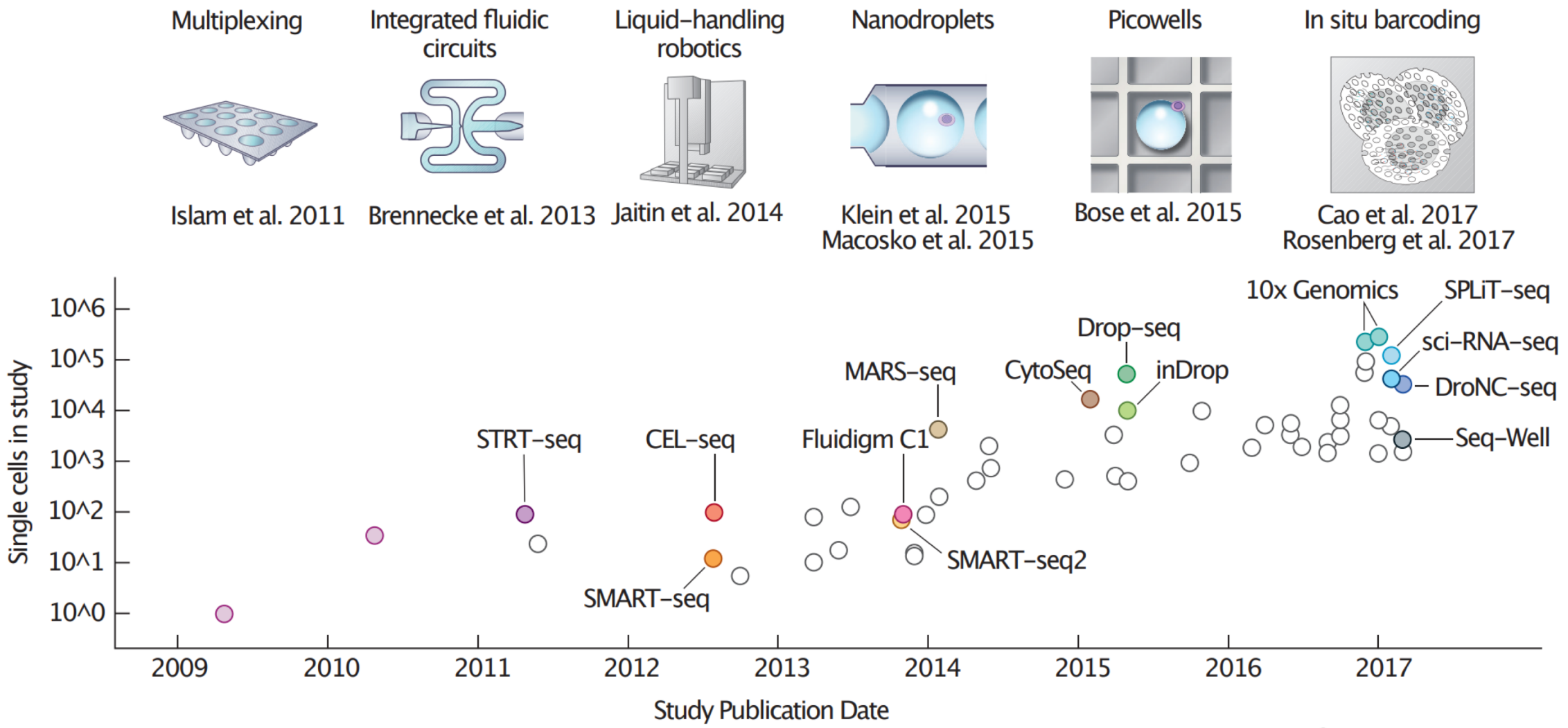
	Cell 1	Cell 2	…
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
…			

Compare gene expression profiles of single cells





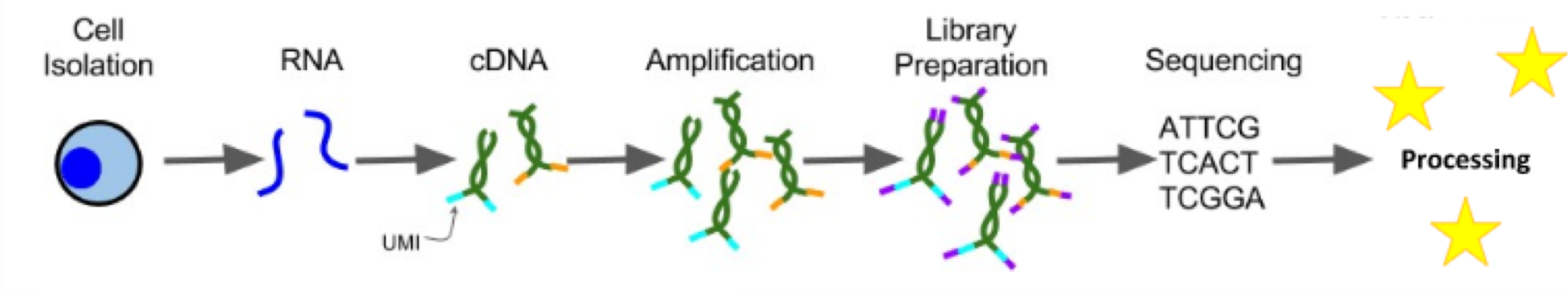
scRNA-seq output has increased significantly



Svensson et al 2018

Challenges in single cell data analysis

Amplification artifacts
Dropouts



Cell Ranger software versions

Chemistry

The assay support of Cell Ranger 3.0, and the previous Cell Ranger 2.2 is summarized below.

Assay Type	Cell Ranger 3.1	Cell Ranger 2.2
Single Cell 3' v3	Yes	No
Single Cell 3' v3 + Feature Barcoding	Yes	No
Single Cell 5'	Yes	Yes
Single Cell 5' + Feature Barcoding	Yes	No
Single Cell 3' v2	Yes	Yes
Single Cell 3' v2 + Feature Barcoding	Yes	No

Gene Expression Algorithms Overview

Genome Alignment

Cell Ranger uses an aligner called **STAR**.
STAR tries to find the longest possible sequence which matches one or more sequences in the reference genome. Because STAR is able to recognize splicing events in this way, it is described as a ‘splice aware’ aligner.

Cell Ranger then uses the transcript annotation GTF to bucket the reads into exonic, intronic, and intergenic, and by whether the reads align (confidently) to the genome.
A read is exonic if at least 50% of it intersects an exon, intronic if it is non-exonic and intersects an intron, and intergenic otherwise.

MAPQ adjustment

For reads that align to a single exonic locus but also align to 1 or more non-exonic loci, the exonic locus is prioritized and the read is considered to be confidently mapped to the exonic locus with MAPQ 255.

Transcriptome Alignment

Cell Ranger further aligns exonic reads to annotated transcripts, looking for compatibility. A read that is compatible with the exons of an annotated transcript, and aligned to the same strand, is considered mapped to the transcriptome. If the read is compatible with a single gene annotation, it is considered uniquely (confidently) mapped to the transcriptome. These confidently mapped reads are the only ones considered for UMI counting.

Cell barcode and UMI filtering

- **Cell barcodes**
  - Must be on static list of known cell barcode sequences
  - May be 1 mismatch away from the list if the mismatch occurs at a low- quality position (the barcode is then corrected).
- **UMIs (Unique Molecular Index)**
  - Must not be a homopolymer, e.g. AAAAAAAAAA
  - Must not contain N
  - Must not contain bases with base quality < 10
  - UMIs that are 1 mismatch away from a higher-count UMI are corrected to that UMI if they share a cell barcode and gene.

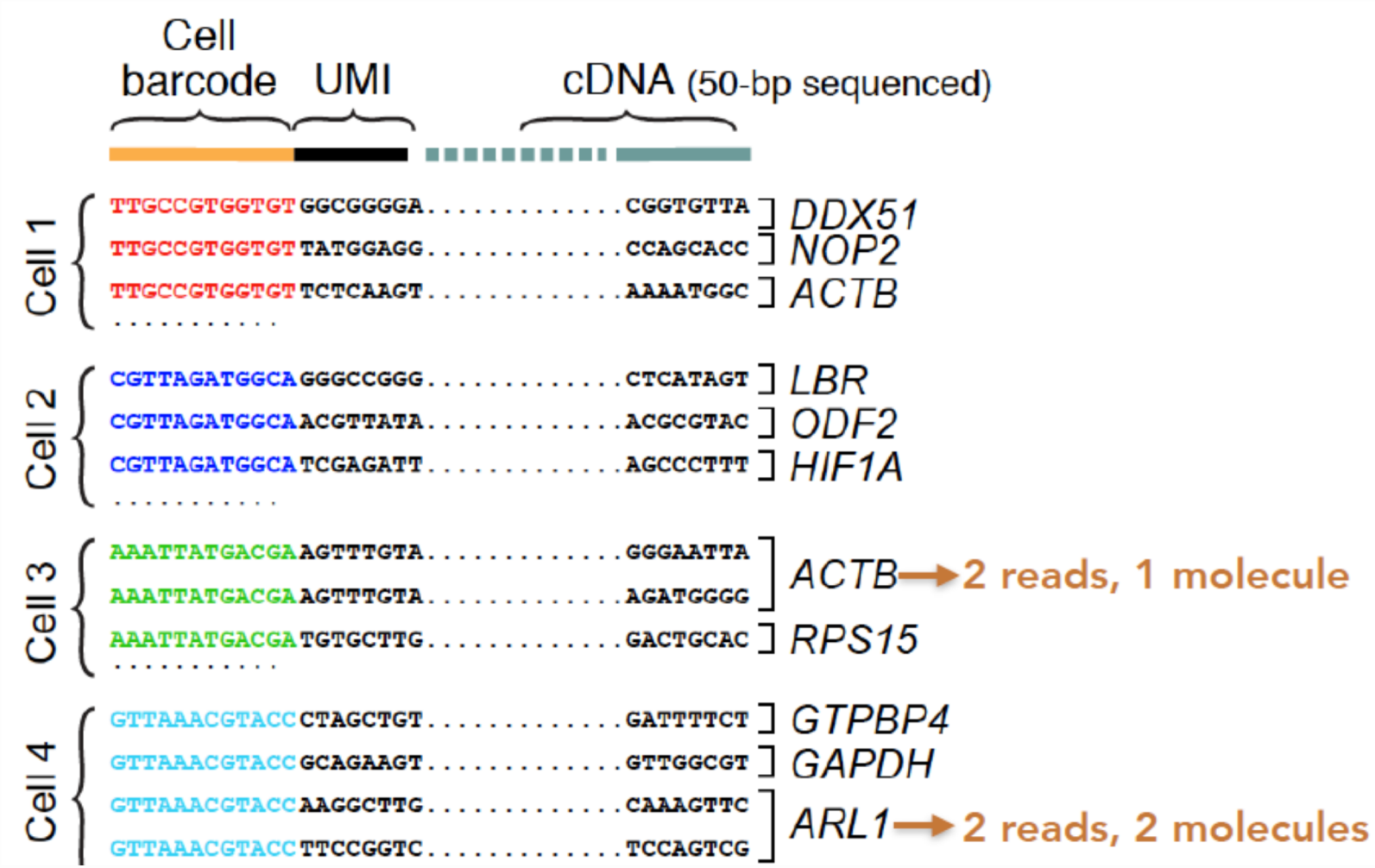
UMI Counting

•Using only the confidently mapped reads with valid barcodes and UMIs,

- Correct the UMIs UMIs are corrected to more abundant UMIs that are one mismatch away in sequence (hamming distance = 1).
- Record which reads are duplicates of the same RNA molecule (PCR duplicates)
- Count only the unique UMIs as unique RNA molecules
- These UMI counts form an unfiltered gene-barcode matrix.

- Cell Ranger again groups the reads by barcode, UMI (possibly corrected), and gene annotation.
- If two or more groups of reads have the same barcode and UMI, but different gene annotations, the gene annotation with the most supporting reads is kept for UMI counting, and the other read groups are discarded.
- In case of a tie for maximal read support, all read groups are discarded, as the gene cannot be confidently assigned.

After these two filtering steps, each observed barcode, UMI, gene combination is recorded as a UMI count in the unfiltered feature-barcode matrix. The number of reads supporting each counted UMI is also recorded in the molecule info file


Cell barcode
UMI
cDNA (50-bp sequenced)
Cell 1
TTGCCGTGGTGTGGCGGGGA.....CGGTGTTA DDX51
TTGCCGTGGTGTTATGGAGG.....CCAGCACC NOP2
TTGCCGTGGTGTTCTCAAGT.....AAAATGGC ACTB
Cell 2
CGTTAGATGGCAGGGCCGGG.....CTCATAGT LBR
CGTTAGATGGCAACGTTATA.....ACGCGTAC ODF2
CGTTAGATGGCATCGAGATT.....AGCCCTTT HIF1A
Cell 3
AAATTATGACGAAGTTTGTA.....GGGAATTA
AAATTATGACGAAGTTTGTA.....AGATGGGG ACTB → 2 reads, 1 molecule
AAATTATGACGATGTGCTTG.....GACTGCAC RPS15
Cell 4
GTTAAACGTACCCTAGCTGT.....GATTTTCT GTPBP4
GTTAAACGTACCGCAGAAGT.....GTTGGCGT GAPDH
GTTAAACGTACCAAGGCTTG.....CAAAGTTC
GTTAAACGTACCTTCCGGTC.....TCCAGTCG ARL1 → 2 reads, 2 molecules

Filtering cells (Cell Ranger)

cellranger 3.0 introduces and improved cell-calling algorithm to identify populations of low RNA content cells, especially when low RNA content cells are mixed into a population of high RNA content cells.
E.g tumor samples often contain large tumor cells mixed with smaller tumor infiltrating lymphocytes (TIL).
The new algorithm is based on the EmptyDrops method (Lun et al., 2018).

The algorithm has two key steps:
1.It uses a cutoff based on total UMI counts of each barcode to identify cells. This step identifies the primary mode of high RNA content cells.

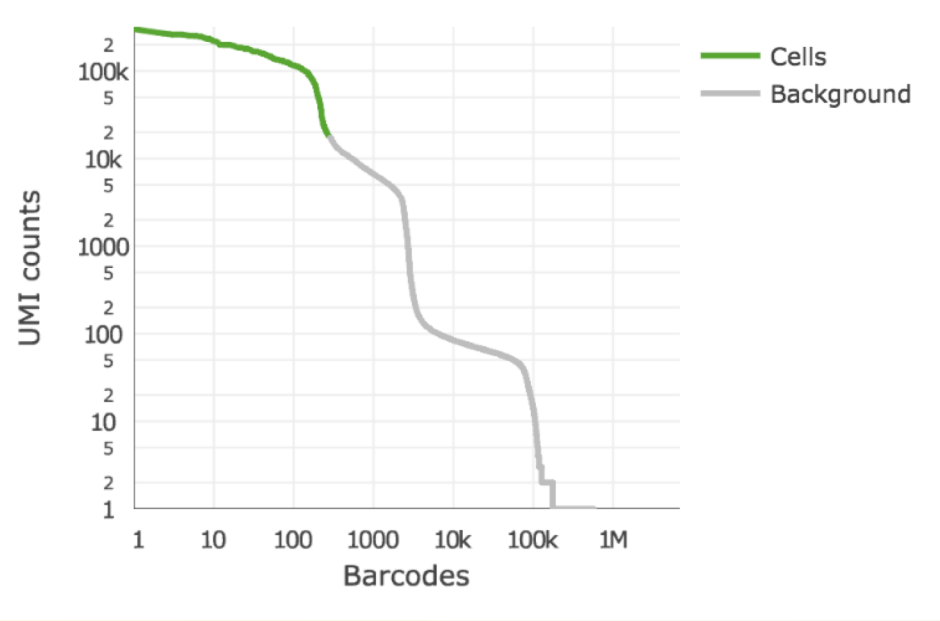
2.Then the algorithm uses the RNA profile of each remaining barcode to determine if it is an “empty” or a cell containing partition. This second step captures low RNA content cells whose total UMI counts may be similar to empty GEMs.

Barcodes selection steps

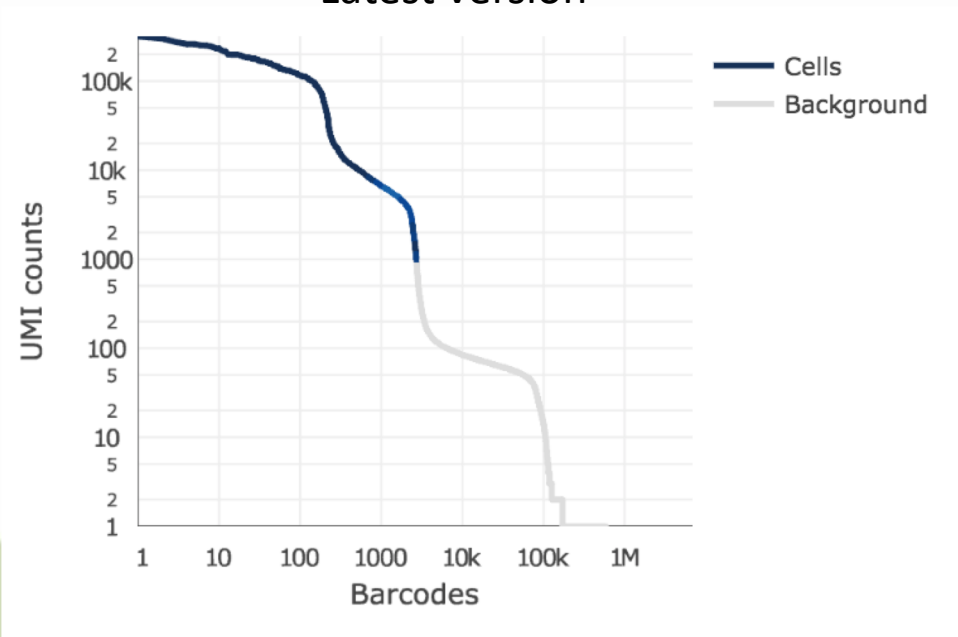
1. The original cellranger cell calling algorithm is used to identify high RNA content cells, using a cutoff based on the total UMI count for each barcode. Let m be the 99th percentile of the top N barcodes by total UMI counts. All barcodes whose total UMI counts exceed m/10 are called as cells in the first pass.

2. In the second step, a set of barcodes with low UMI counts that likely represent ‘empty’ GEM partitions is selected. A model of the RNA profile of selected barcodes is created. This model, called the background model, provides a non-zero estimate for genes that were not observed in the 1st set. Barcodes whose RNA profile strongly disagrees with the background model are “rescued”. This second step identifies cells that are clearly distinguishable from the profile of empty GEMs, even though they may have much lower RNA content than the largest cells in the experiment.

Earlier version



Latest version



Pipeline download and installation

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/using/tutorial_in

Command-Line Argument Reference

Argument	Description
--id	A unique run ID string: e.g. sample345
--fastqs	:Path of the fastq_path folder generated by cellranger mkfastq e.g. /home/jdoe/runs/HAWT7ADXX/outs/fastq_path. This contains a directory hierarchy that cellranger count will automatically traverse. - OR - Any folder containing fastq files, for example if the fastq files were generated by a service provider and delivered outside the context of the mkfastq output directory structure. Can take multiple comma-separated paths
--libraries	Path to a libraries.csv file declaring FASTQ paths and library types of input libraries. Required for feature-barcoding analysis. See Feature Barcoding Analysis page for details. When using this argument, --fastqs and --sample must not be passed.
--sample	Sample name as specified in the sample sheet supplied to cellranger mkfastq. Can take multiple comma-separated values,
--transcriptome	•Path to the Cell Ranger compatible transcriptome reference e.g.For a human-only sample, use /opt/refdata-cellranger-GRCh38-3.0.0 •For a human and mouse mixture sample, use /opt/refdata-cellranger-hg19-and-mm10-3.0.0
--feature-ref	Path to a Feature Reference CSV file declaring the Feature Barcoding reagents in use in the experiment. Required for Feature Barcoding analysis. See Feature Barcode Reference for details on how to construct the feature reference.
--expect-cells	(optional) Expected number of recovered cells. Default: 3,000 cells.
--force-cells	(optional) Force pipeline to use this number of cells, bypassing the cell detection algorithm.
--chemistry	•(optional)
--r1-length	(optional)
--r2-length	(optional) Hard-trim the input R2 sequence to this length.
--localcores	Restricts cellranger to use specified number of cores to execute pipeline stages. By default, cellranger will use all of the cores available on your system.
--localmem	Restricts cellranger to use specified amount of memory (in GB) to execute pipeline stages. By default, cellranger will use 90% of the memory available on your system.

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/using/count

How to run cellranger

```
path_to_cellranger/cellranger count --id=samplename(your choice) --transcriptome=/opt/refdata-cellranger-GRCh38-3.0.0  --fastqs=/home/jdoe/runs/HAWT7ADXX/outs/fastq_path  --sample=sample_prefix
```

Output

The output of the pipeline will be contained in a folder named with the sample ID you specified (e.g. sample345). The subfolder named outs will contain the main pipeline output files:
Once cellranger count has successfully completed, you can browse the resulting summary HTML file in any supported web browser, open the .cloupe file in Loupe Cell Browser, or refer to the Understanding Output section to explore the data by hand.

Matrix output

With 3 files needed to completely describe each gene x cell matrix
•matrix.mtx.gz
•features.tsv.gz
•barcode.tsv.gz

Type	Description
Raw	gene-barcode matrices Contains every barcode from fixed list of known-good barcode sequences. This includes background and non-cellular barcodes.
Filtered	gene-barcode matrices Contains only detected cellular barcodes.

Bam output

10x Chromium cellular and molecular barcode information for each read is stored as TAG fields:
The following TAG fields are present if a read maps to the genome **and** overlaps an exon by at least one base pair.
A read may align to multiple transcripts and genes, but it is only considered confidently mapped to the transcriptome it if mapped to a single gene.

Tag	Description
CB	Chromium cellular barcode sequence that is error-corrected and confirmed against a list of known-good barcode sequences.
CR	Chromium cellular barcode sequence as reported by the sequencer.
CY	Chromium cellular barcode read quality. Phred scores as reported by sequencer.
UB	Chromium molecular barcode sequence that is error-corrected among other molecular barcodes with the same cellular barcode and gene alignment.
UR	Chromium molecular barcode sequence as reported by the sequencer.
UY	Chromium molecular barcode read quality. Phred scores as reported by sequencer.
BC	Sample index read.
QT	Sample index read quality. Phred scores as reported by sequencer.

10X genomics sample report

Summary of the alignment and assignment of reads to cells and genes are present in the metrics_summary.csv.

Metric	Description
Valid Barcodes	Fraction of reads with cell-barcodes that match the whitelist.
Reads Mapped Confidently to Transcriptome	Fraction of reads that mapped to a unique gene in the transcriptome with a high mapping quality score as reported by the aligner.
Reads Mapped Confidently to Exonic Regions	Fraction of reads that mapped to the exonic regions of the genome with a high mapping quality score as reported by the aligner.
Reads Mapped Antisense to Gene	Fraction of reads confidently mapped to the transcriptome, but on the opposite strand of their annotated gene. A read is counted as antisense if it has any alignments that are consistent with an exon of a transcript but antisense to it, and has no sense alignments.
Sequencing Saturation	The fraction of reads originating from an already-observed UMI. This is a function of library complexity and sequencing depth. More specifically, this is the fraction of confidently mapped, valid cell-barcode, valid UMI reads that had a non-unique (cell-barcode, UMI, gene).
Fraction Reads in Cells	The fraction of cell-barcoded, confidently mapped reads with cell-associated barcodes.
Total Genes Detected	The number of genes with at least one UMI count in any cell.
Median UMI Counts per Cell	The median number of total UMI counts across all cell-associated barcodes.

Output examples

Sample_1

Summary | Analysis

9,175
Estimated Number of Cells

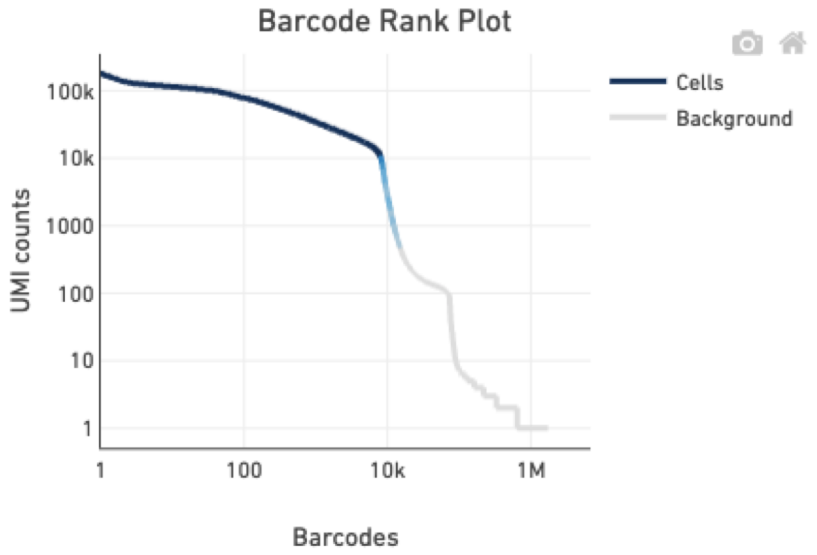
48,398
Mean Reads per Cell

3,976
Median Genes per Cell

Sequencing

Number of Reads	444,047,625
Valid Barcodes	95.0%
Valid UMIs	99.9%
Sequencing Saturation	24.4%
Q30 Bases in Barcode	95.0%
Q30 Bases in RNA Read	93.2%
Q30 Bases in Sample Index	92.8%
Q30 Bases in UMI	92.9%

Cells



Estimated Number of Cells	9,175
Fraction Reads in Cells	88.5%
Mean Reads per Cell	48,398
Median Genes per Cell	3,976
Total Genes Detected	21,633
Median UMI Counts per Cell	17,519

Mapping

Reads Mapped to Genome	90.9%
Reads Mapped Confidently to Genome	84.6%
Reads Mapped Confidently to Intergenic Regions	3.5%
Reads Mapped Confidently to Intronic Regions	12.1%
Reads Mapped Confidently to Exonic Regions	69.1%
Reads Mapped Confidently to Transcriptome	65.7%
Reads Mapped Antisense to Gene	1.6%

Sample

Sample ID	Sample_1
Sample Description	
Chemistry	Single Cell 3' v3
Transcriptome	mm10_eGFP-
Pipeline Version	3.1.0

Sample_2

5,663
Estimated Number of Cells

38,427
Mean Reads per Cell

1,981
Median Genes per Cell

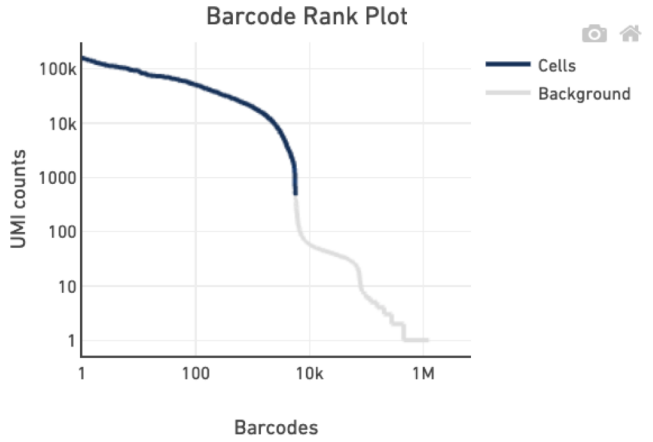
Sequencing

Number of Reads	217,612,940
Valid Barcodes	95.8%
Valid UMIs	99.9%
Sequencing Saturation	55.4%
Q30 Bases in Barcode	94.8%
Q30 Bases in RNA Read	91.1%
Q30 Bases in Sample Index	93.4%
Q30 Bases in UMI	92.2%

Mapping

Reads Mapped to Genome	94.9%
Reads Mapped Confidently to Genome	92.1%
Reads Mapped Confidently to Intergenic Regions	2.8%
Reads Mapped Confidently to Intronic Regions	9.5%
Reads Mapped Confidently to Exonic Regions	79.8%
Reads Mapped Confidently to Transcriptome	75.7%
Reads Mapped Antisense to Gene	1.1%

Cells



Estimated Number of Cells	5,663
Fraction Reads in Cells	94.1%
Mean Reads per Cell	38,427
Median Genes per Cell	1,981
Total Genes Detected	17,517
Median UMI Counts per Cell	7,938

Sample

Sample ID	Sample_1_count
Sample Description	
Chemistry	Single Cell 3' v3
Transcriptome	mm10-3.0.0
Pipeline Version	3.1.0

Troubleshooting

The analysis detected some issues. Details »

Alert	Value	Detail
⚠ **Low Fraction Reads Confidently Mapped To Transcriptome**	24.4%	Ideal > 30%. This can indicate use of the wrong reference transcriptome, a reference transcriptome with overlapping genes, poor library quality, poor sequencing quality, or reads shorter than the recommended minimum. Application performance may be affected.

Estimated Number of Cells

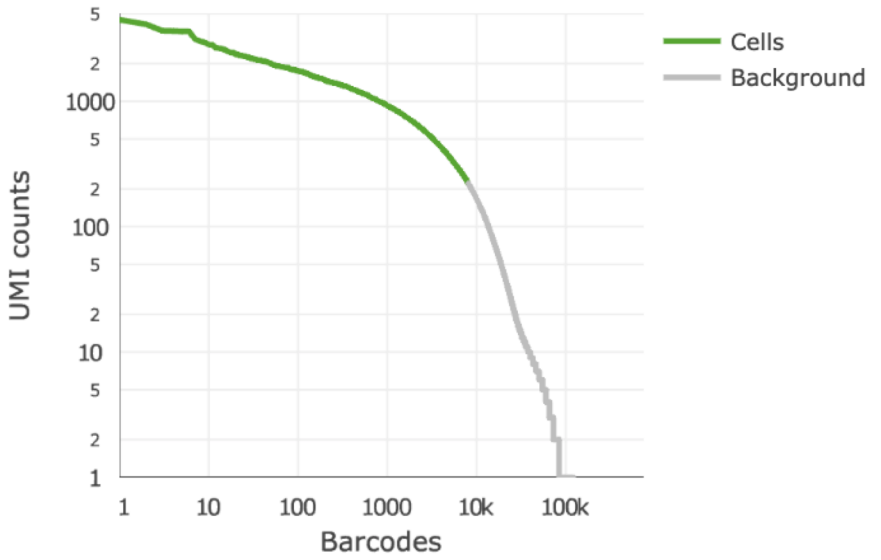
8,109

Mean Reads per Cell	Median Genes per Cell
46,965	**334**

Sequencing

Number of Reads	380,844,038
Valid Barcodes	82.0%
Sequencing Saturation	92.8%
Q30 Bases in Barcode	95.8%
Q30 Bases in RNA Read	82.9%
Q30 Bases in Sample Index	91.4%
Q30 Bases in UMI	95.2%

Cells



Estimated Number of Cells	8,109
Fraction Reads in Cells	74.8%
Mean Reads per Cell	46,965
Median Genes per Cell	334
Total Genes Detected	25,426
Median UMI Counts per Cell	426

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	Pool_1_S1_L001_R2_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	380844038
Sequences flagged as poor quality	0
Sequence length	150
%GC	36

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTT	103227946	27.105044506433888	No Hit
GCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTT	8592717	2.256229884843307	No Hit
AGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTT	4198600	1.1024460359282295	No Hit

Single Cell 5' assay after reverse transcription:

The analysis detected some issues. Details »

Alert	Value	Detail
⚠ Low Fraction Reads in Cells	53.4%	Ideal > 70%. Application performance may be affected. Many of the reads were not assigned to cell-associated barcodes. This could be caused by high levels of ambient RNA or by a significant population of cells with a low RNA content, which the algorithm did not call as cells. The latter case can be addressed by inspecting the data to determine the appropriate cell count and using --force-cells.

Estimated Number of Cells

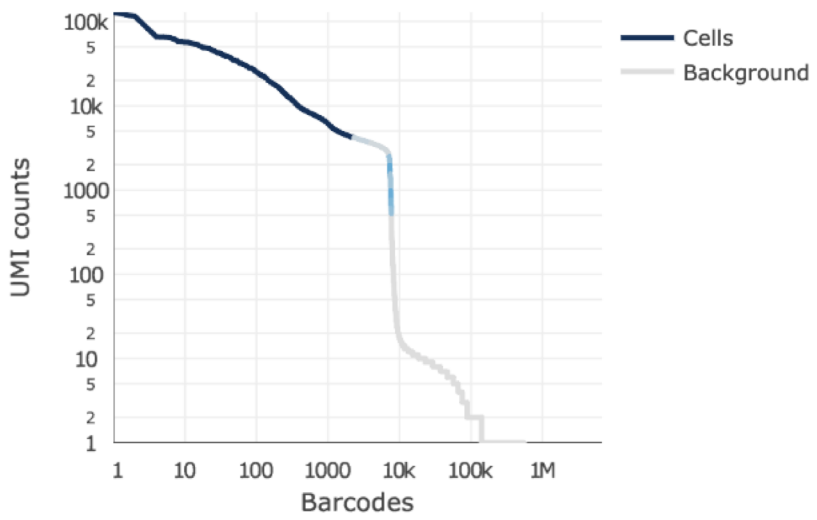
2,553

Mean Reads per Cell	Median Genes per Cell
130,565	2,064

Sequencing

Number of Reads	333,334,060
Valid Barcodes	97.5%
Sequencing Saturation	74.3%
Q30 Bases in Barcode	96.1%
Q30 Bases in RNA Read	90.8%
Q30 Bases in Sample Index	94.8%
Q30 Bases in UMI	93.3%

Cells



Estimated Number of Cells	2,553
Fraction Reads in Cells	53.4%
Mean Reads per Cell	130,565
Median Genes per Cell	2,064
Total Genes Detected	18,966
Median UMI Counts per Cell	5,148

Estimated Number of Cells

8,000

Mean Reads per Cell	Median Genes per Cell
41,666	1,586

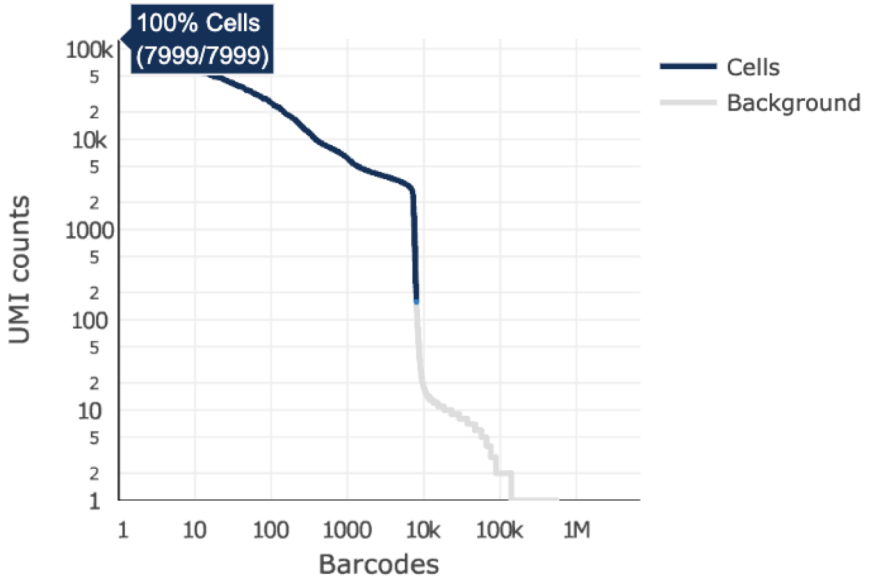
Sequencing

Number of Reads	333,334,060
Valid Barcodes	97.5%
Sequencing Saturation	74.3%
Q30 Bases in Barcode	96.1%
Q30 Bases in RNA Read	90.8%
Q30 Bases in Sample Index	94.8%
Q30 Bases in UMI	93.3%

Mapping

Reads Mapped to Genome	94.9%
Reads Mapped Confidently to Genome	93.0%
Reads Mapped Confidently to Intergenic Regions	5.0%
Reads Mapped Confidently to Intronic Regions	38.4%
Reads Mapped Confidently to Exonic Regions	49.6%
Reads Mapped Confidently to Transcriptome	45.7%
Reads Mapped Antisense to Gene	1.4%

Cells



Estimated Number of Cells	8,000
Fraction Reads in Cells	97.9%
Mean Reads per Cell	41,666
Median Genes per Cell	1,586
Total Genes Detected	19,782
Median UMI Counts per Cell	3,614

Sample

Name	_GEX_count_8k
Description	
Transcriptome	GRCh38
Chemistry	Single Cell 3' v3
Cell Ranger Version	3.0.0

Conclusion

Always perform QC of your libraries!

Be aware of library specifics → critical mindset!