Single Cell Transcriptomics

From Experimental Design to Data Analysis

Day 1

University of Maryland
Baltimore County

Workshop Outline

Part I: Background & Technology Overview

Part II: Sample Preparation, Submission, QC, and Sequencing

Part III: Workshop - Using the Linux/Unix command line

Part IV: Data Downloading and Preprocessing

Goals of this Workshop

Today

- Gain familiarity with current single cell technologies and use cases
- Learn best practices for single cell sample generation and handling
- Learn common pitfalls in single cell experiments and how to avoid them
- Understand how to ensure that your samples are high-quality
- Get comfortable using the linux command line
- Download raw sequencing data, demultiplex, and generate DGE matrix files

Overall

- Learn how to design and perform a “full stack” single cell RNA-seq experiment

Workshop Outline – Part I

Part I: Background & Technology Overview

Why use single cell?

How does single cell technology work?

Overview of single cell methods

Focus on: 10X Genomics single cell gene expression profiling

Available single cell technology variations/extensions

- scATAC-seq
- CITE-seq
- Spatial transcriptomics

Workshop Outline – Part II

Part II: Sample Preparation, Submission, QC, and Sequencing

Whole cell dissociation, nucleus extraction, and FACS

Assessing sample purity and QC

Sample submission steps

- Core facility selection & sample delivery
- cDNA/Sequencing Library QC metrics

Sequencing technologies & use cases

- How are libraries read on the sequencer?
- What sequencing platforms are compatible with scRNA-seq?

Getting back data

- Timeline and expectations

Workshop Outline – Part IV

Part III: Workshop

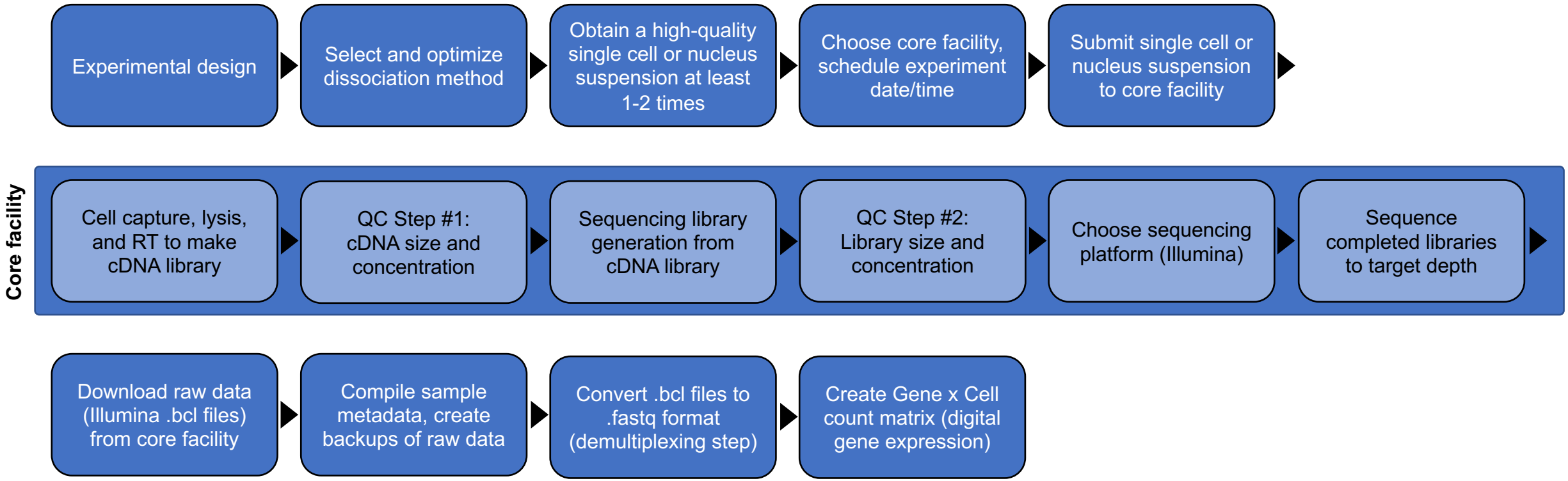
Using the Linux/Unix command line

Workshop Outline – Part III

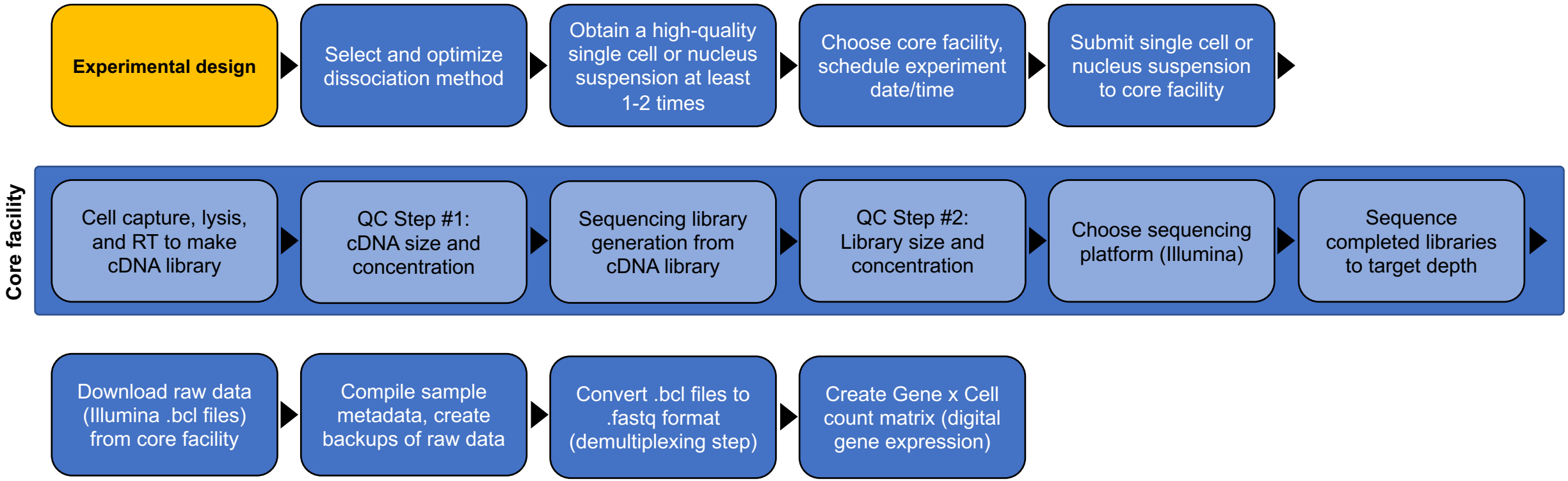
Part IV: Data Downloading and Preprocessing

- Introduction to HPCC
- How to download data from a sequencing core
- Data formats, storage, and preprocessing
- Generating fastq files with Cellranger (bcl2fastq)
- Generating DGE matrices from fastq files with Cellranger

Single Cell Experiment Outline

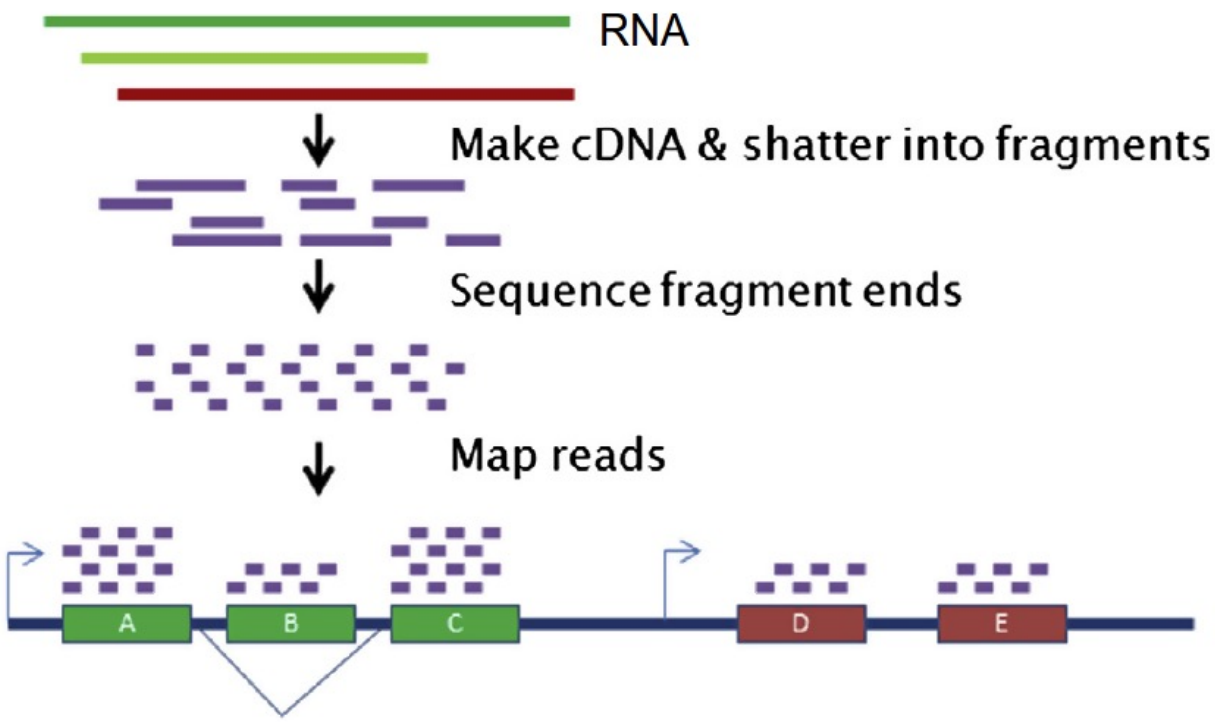


Single Cell Experiment Outline

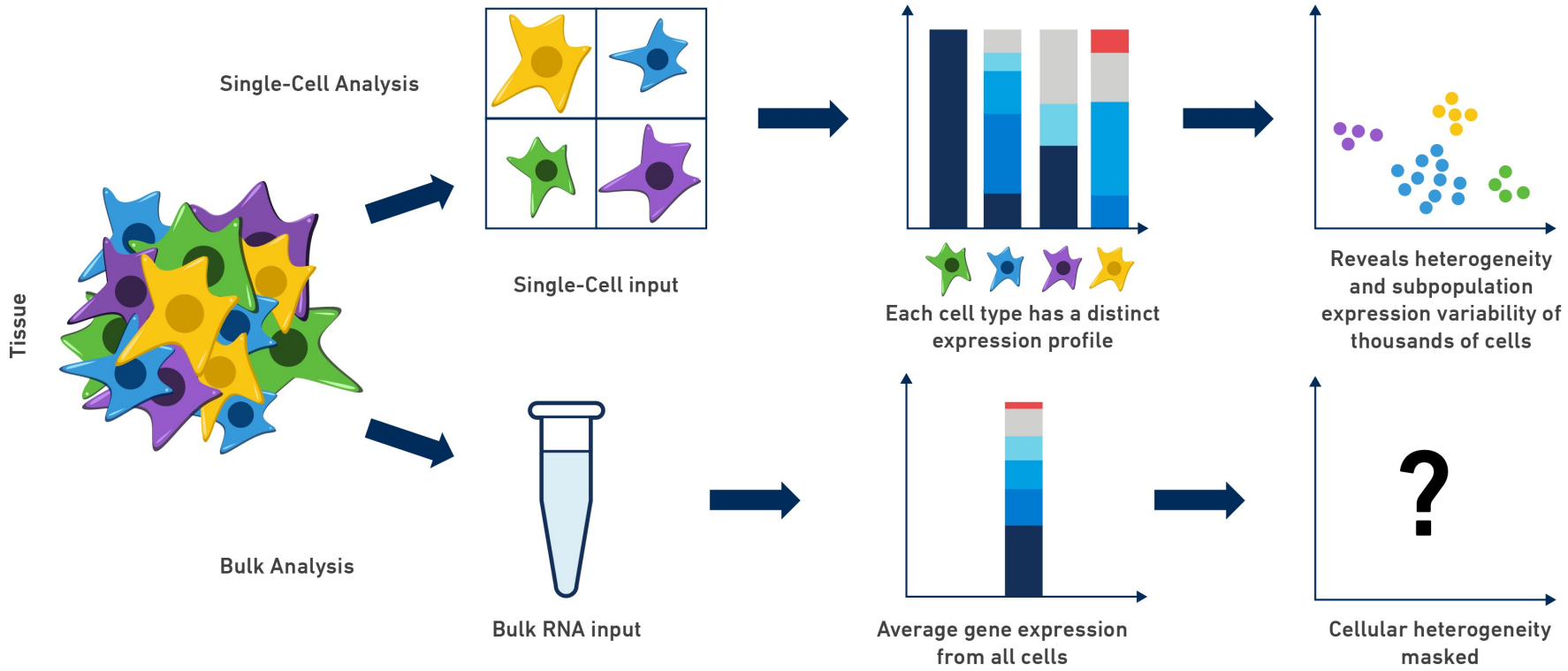


Part I: Background & Technology Overview

Basic outline of RNA Sequencing AKA 'Transcriptomics'

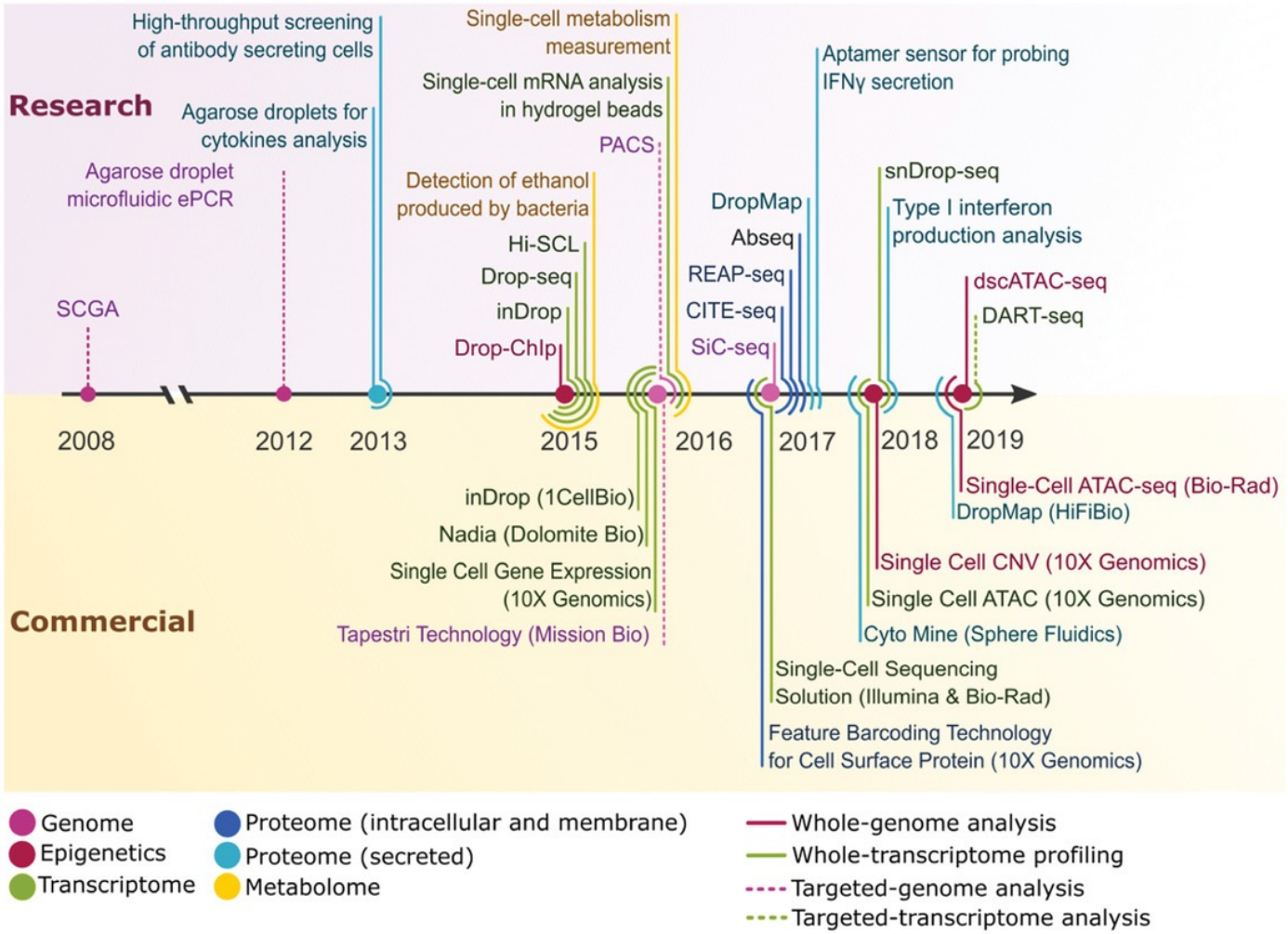


Why use single cell RNA-seq?



[Source: 10X Genomics Blog]

Timeline of scRNA-seq Technologies



[Matuła et al., 2019 *Advanced Biosystems*]

How does scRNA-seq technology work?

List [edit]

Method	Reference	Sequencing Mode	Early Estimate	Late Estimate
Tang method	[2]	Short Reads	2008	2009
CyTOF	[3]	Short Reads	2011	2012
STRT-seq / C1	[4]	Short Reads	2011	2012
SMART-seq	[5]	Short Reads	2012	2013
CEL-seq	[6]	Short Reads	2012	2013
Quartz-Seq	[7]	Short Reads	2012	2013
PMA / SMA	[8]	Short Reads	2012	2013
scBS-seq	[9]	Short Reads	2013	2014
AbPair	[10]	Short Reads	2014	2014
MARS-seq	[11]	Short Reads	2014	2015
DR-seq	[12]	Short Reads	2014	2015
G&T-Seq	[13]	Short Reads	2014	2015
SCTG	[14]	Short Reads	2014	2015
SIDR-seq	[15]	Short Reads	2014	2015
sci-ATAC-seq	[16]	Short Reads	2014	2015
Hi-SCL	[17]	Short Reads	2015	2015
SUPeR-seq	[18]	Short Reads	2015	2015
Drop-Chip	[19]	Short Reads	2015	2015
CytoSeq	[20]	Short Reads	2015	2016
inDrop	[21]	Short Reads	2015	2016
sc-GEM	[22]	Short Reads	2015	2016
scTrio-seq	[23]	Short Reads	2015	2016
scM&T-seq	[24]	Short Reads	2015	2016
PLAYR	[25]	Short Reads	2015	2016
Genshaft-et-al-2016	[26]	Short Reads	2015	2016
Darmanis-et-al-2016	[27]	Short Reads	2015	2016
CRISP-seq	[28]	Short Reads	2015	2016
scGESTALT	[29]	Short Reads	2015	2016
CEL-Seq2 / C1	[30]	Short Reads	2015	2016
STRT-seq-2i	[31]	Short Reads	2016	2017
RNAseq @10xgenomics	[32]	Short Reads	2016	2017
RNAseq / Gene Expression @nanostringtech	[33]	Short Reads	2016	2017

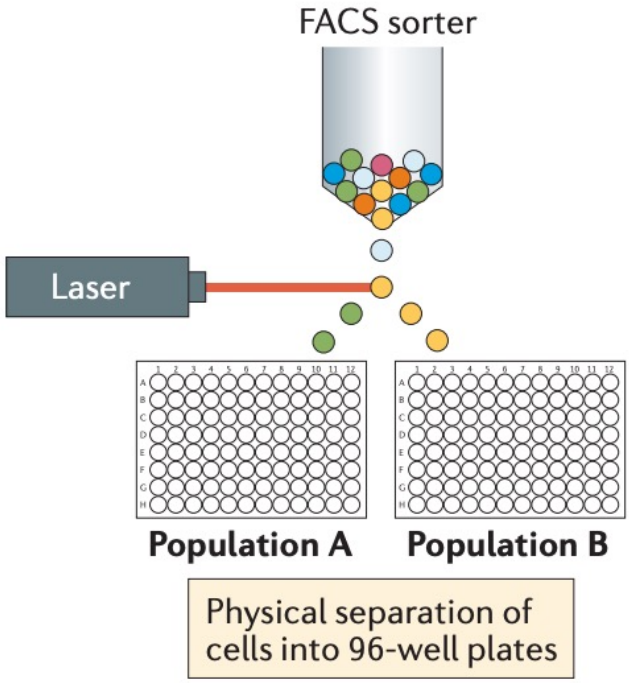
>100 single cell sequencing methods have been published in the last decade

[https://en.wikipedia.org/wiki/List_of_single_cell_omics_methods]

How does scRNA-seq technology work?

Plate-Based

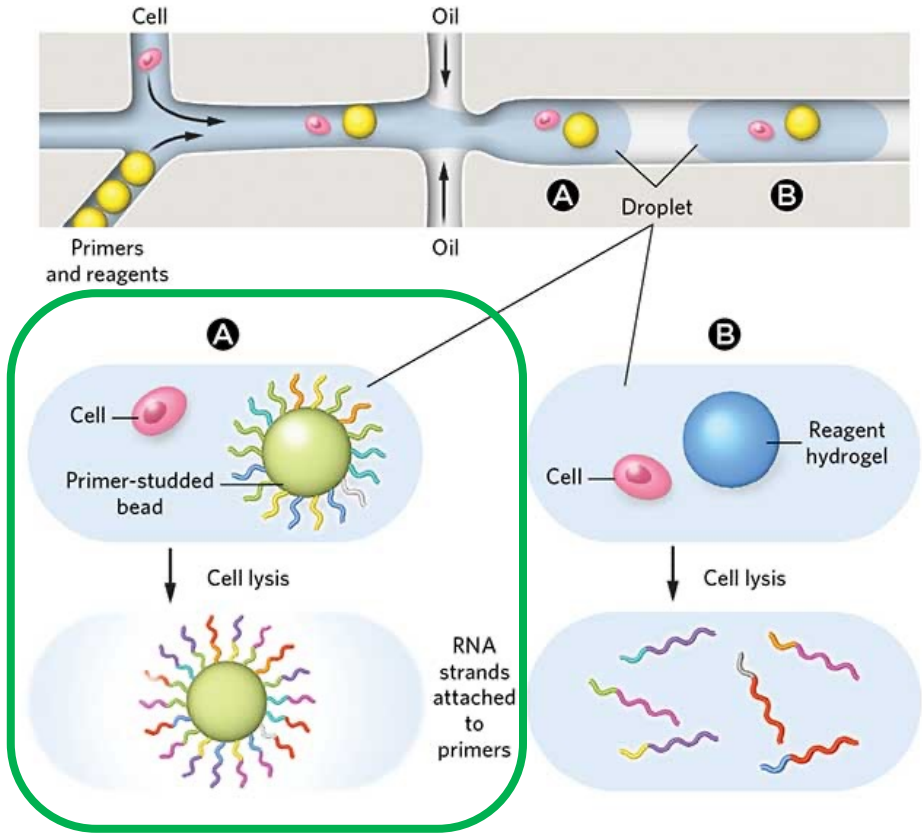
Low Throughput, High Depth, High cost/cell
Low input requirement (good for rare cell types)
Total RNA



"Hybrid" option: use FACS to enrich cell population of interest, and use sorted cells as input to droplet-based methods

Droplet-Based

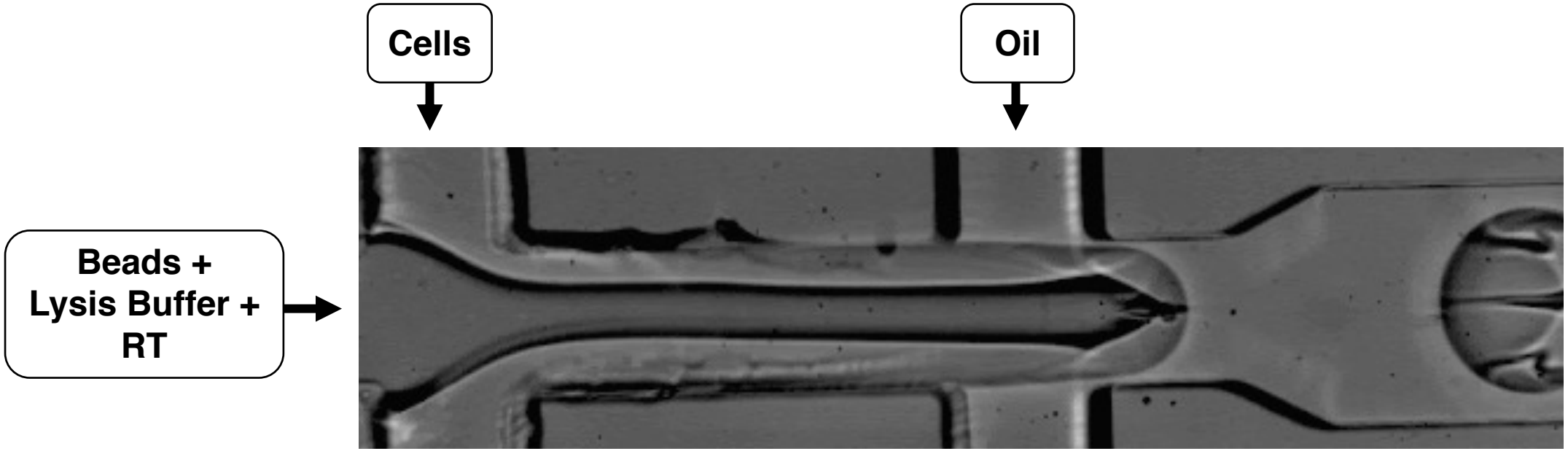
High Throughput, Low Depth, Low cost/cell
Medium-high input requirement (not great for very rare cell types)
mRNA only



A: Dropseq/10X
B: inDrop

[Source: The Scientist Magazine]

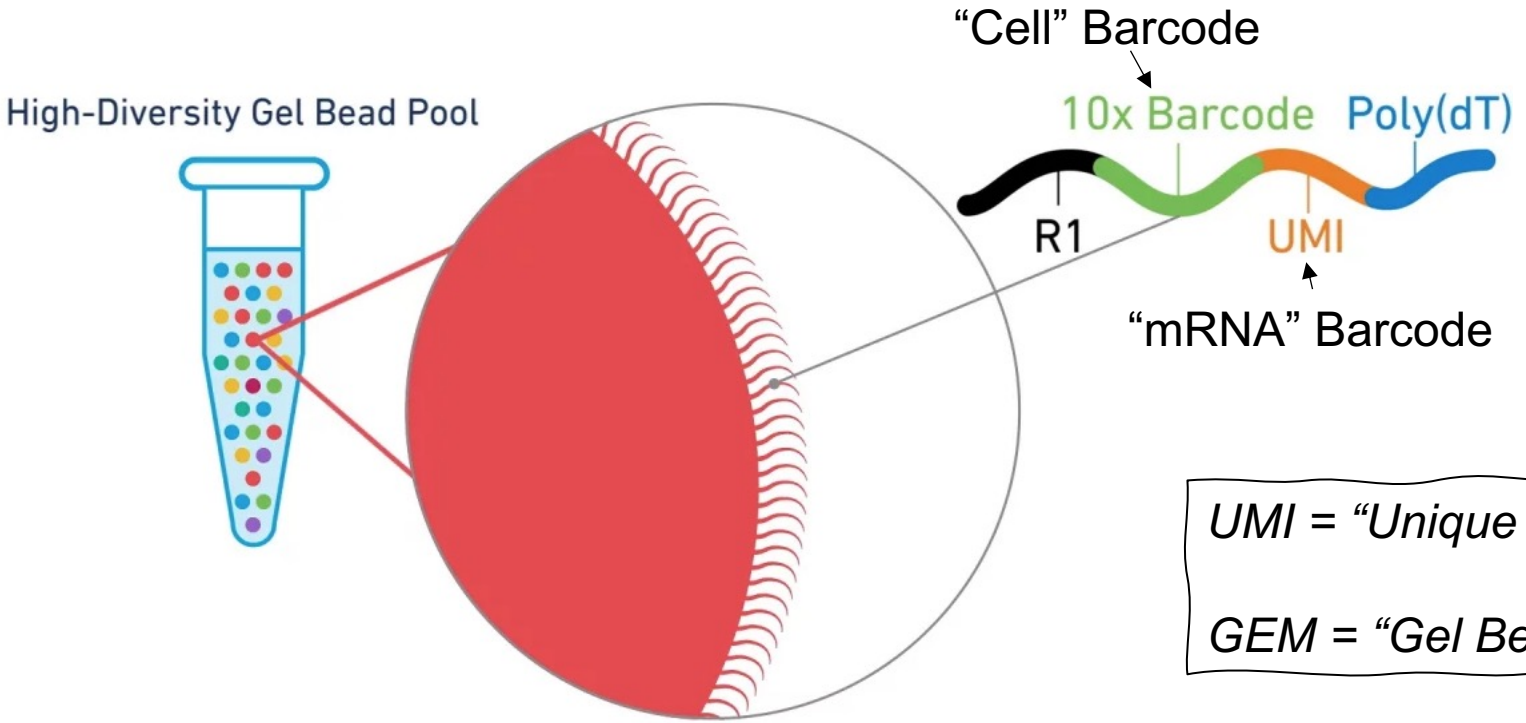
Rapid microfluidic droplet generation partitions cells into isolated “reaction chambers”



[Source: The McCarroll Lab]

Transcriptome capture and barcoding by small oligo-coupled beads

Single cell transcriptome capture and UMI barcoding

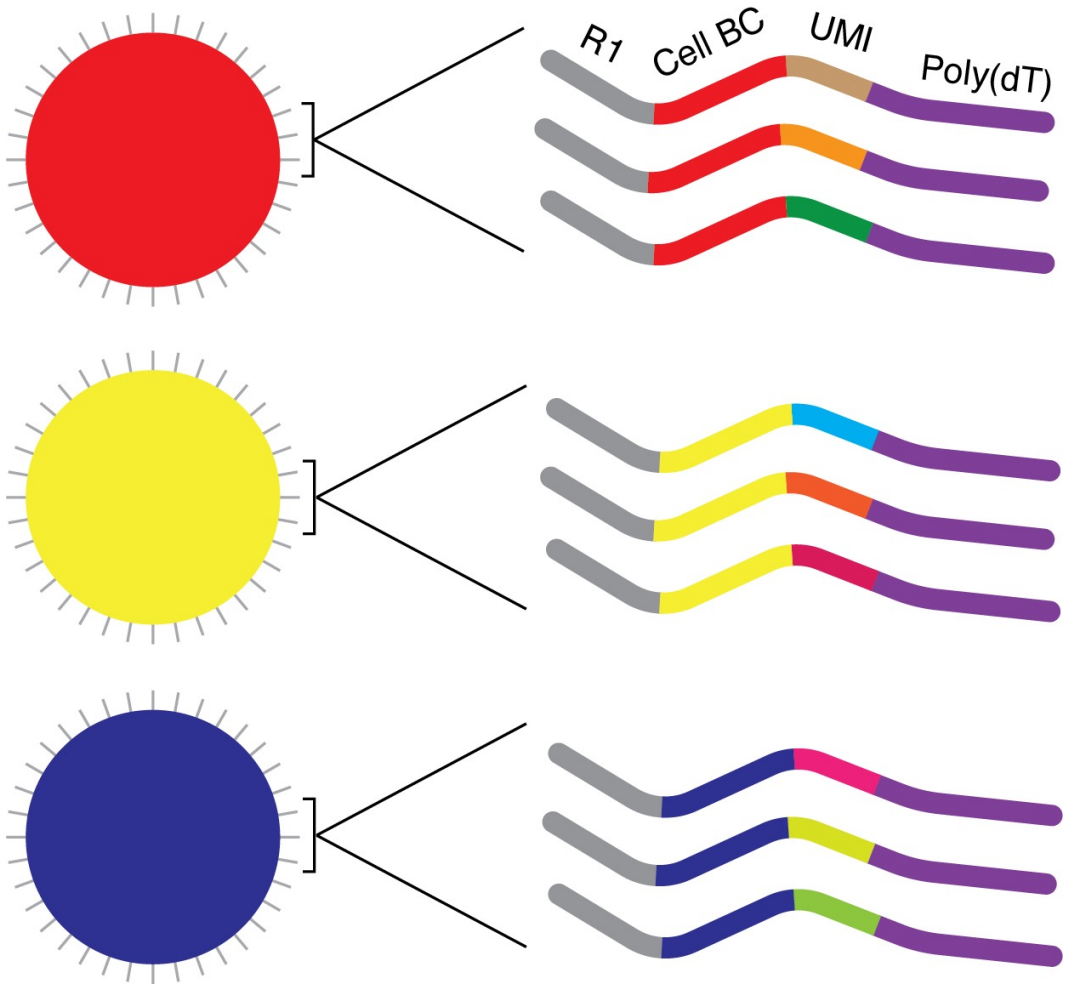


UMI = “Unique Molecular Identifier”

GEM = “Gel Bead-in-Emulsion” (10X)

[Source: 10X Genomics]

Transcriptome capture and barcoding by small oligo-coupled beads



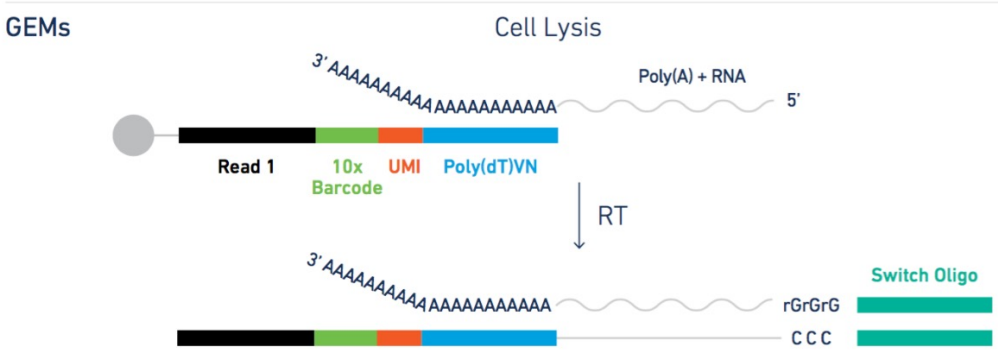
R1: identical on all beads and oligos

Cell Barcode (16bp): unique to each bead, identical on all oligos of a single bead

UMI (12bp): unique to each oligo on all beads

Poly(dT) identical on all beads and oligos

How does scRNA-seq technology work?



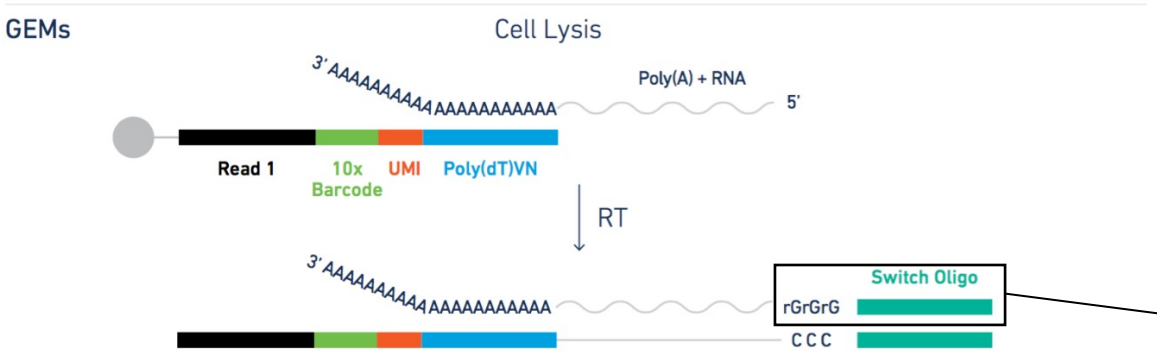
"Reverse transcriptases derived from Moloney Murine Leukemia Virus (MMLV) have an intrinsic terminal transferase activity, which causes the addition of a few non-templated nucleotides at the 3´ end of cDNA, with a preference for cytosine. This mechanism can be exploited to make the reverse transcriptase switch template from the RNA molecule to a secondary oligonucleotide during first-strand cDNA synthesis, and thereby to introduce arbitrary barcode or adaptor sequences in the cDNA."

UMI = "Unique Molecular Identifier"

GEM = "Gel Bead-in-Emulsion (10X)

[Source: 10X Genomics
Zajac et al., 2013 *PLOS One*]

How does scRNA-seq technology work?



"Reverse transcriptases derived from Moloney Murine Leukemia Virus (MMLV) have an intrinsic terminal transferase activity, which causes the addition of a few non-templated nucleotides at the 3´ end of cDNA, with a preference for cytosine. This mechanism can be exploited to make the reverse transcriptase switch template from the RNA molecule to a secondary oligonucleotide during first-strand cDNA synthesis, and thereby to introduce arbitrary barcode or adaptor sequences in the cDNA."

Question: What is a template switch oligo (TSO)?

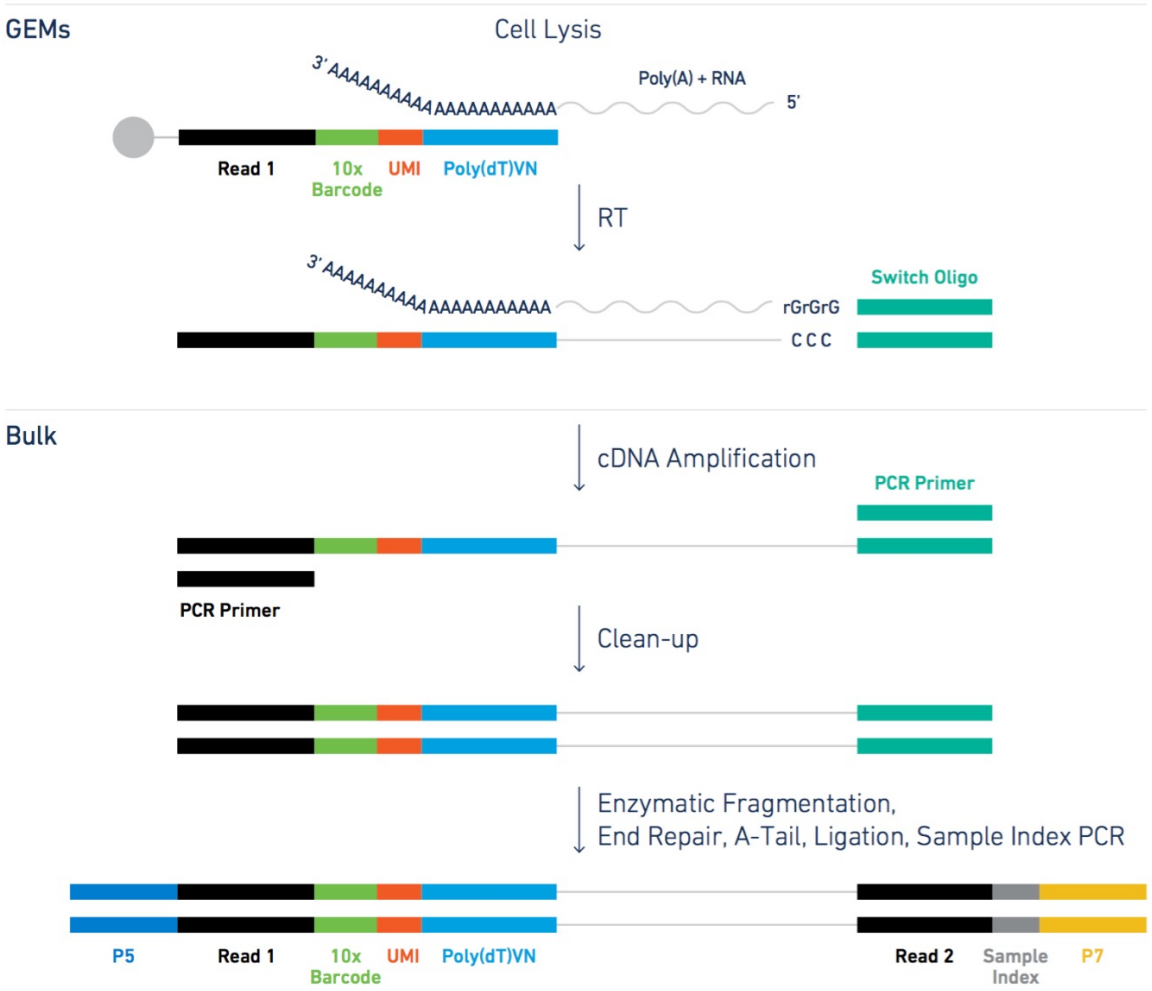
Answer: The TSO (template switch oligo) is an oligo that hybridizes to untemplated C nucleotides added by the reverse transcriptase during reverse transcription. The TSO adds a common 5' sequence to full length cDNA that is used for downstream cDNA amplification.

UMI = "Unique Molecular Identifier"

GEM = "Gel Bead-in-Emulsion (10X)

[Source: 10X Genomics]

How does scRNA-seq technology work?



UMI = "Unique Molecular Identifier"

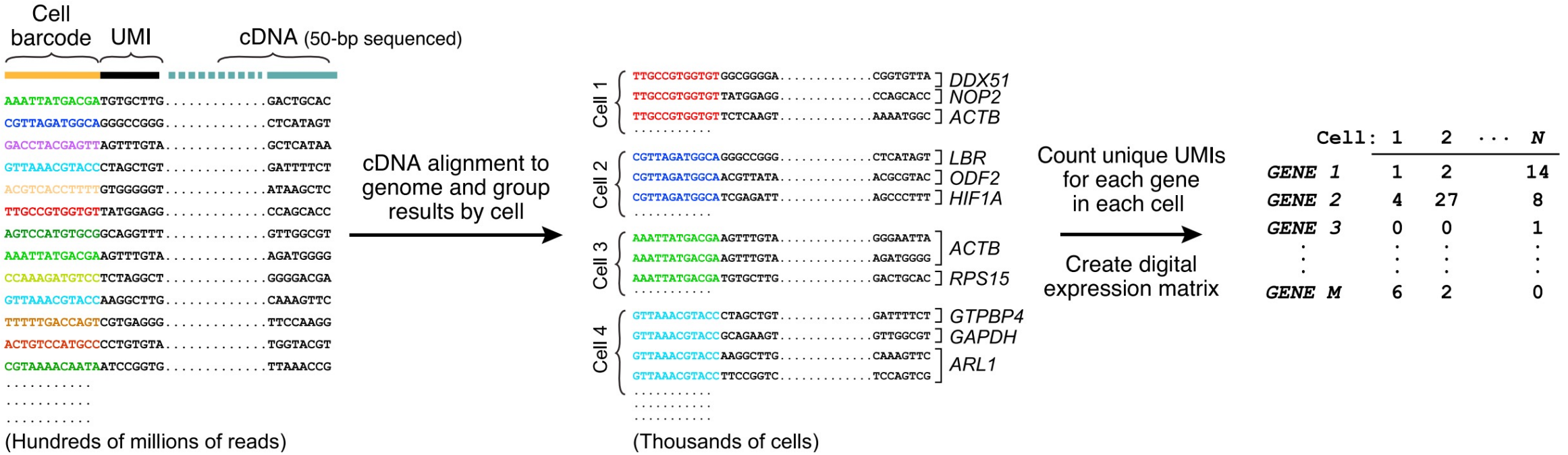
GEM = "Gel Bead-in-Emulsion (10X)

[Source: 10X Genomics]

How does scRNA-seq technology work?

[Source: 10X Genomics]

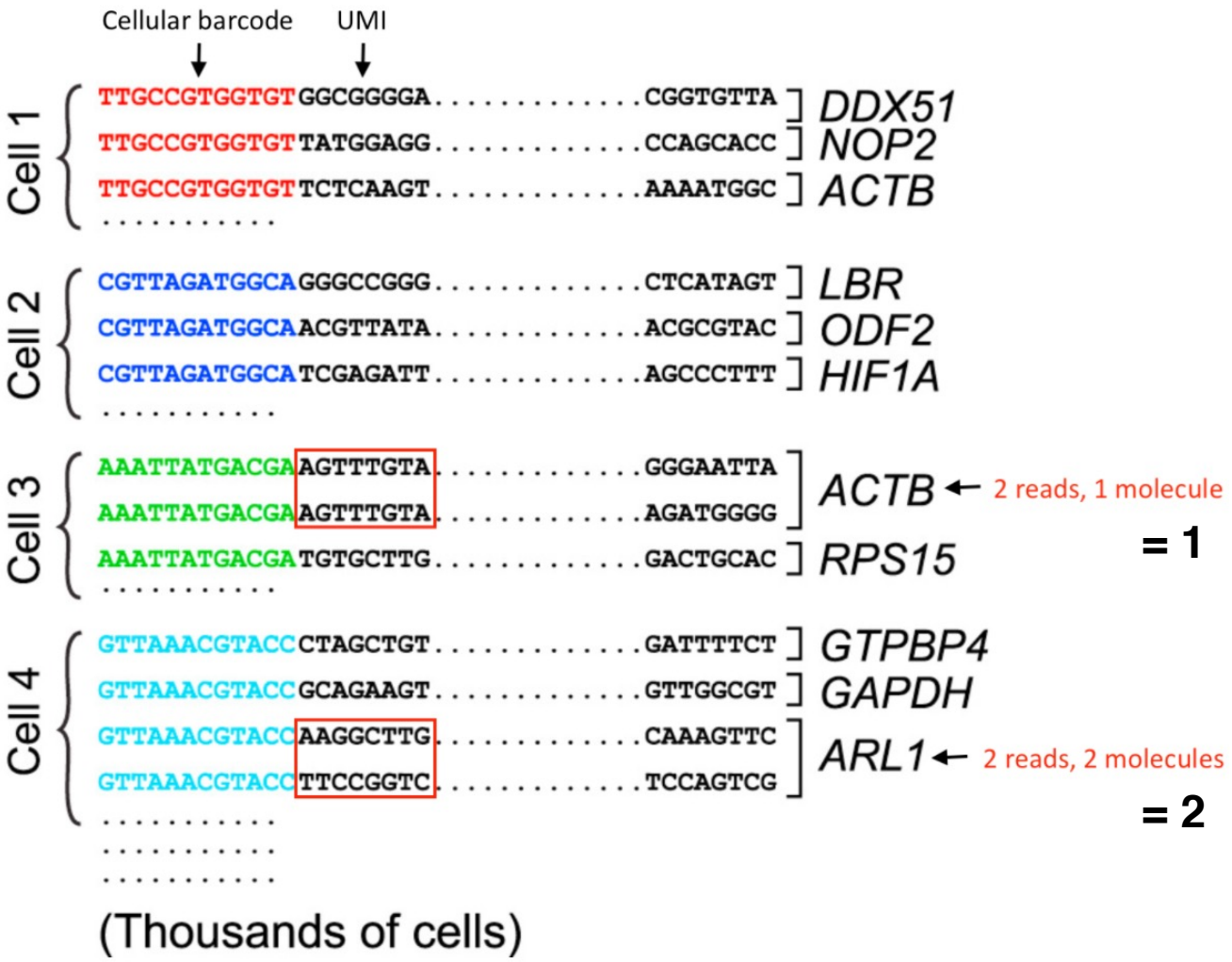
How does scRNA-seq technology work?



DGE = "Digital Gene Expression"

[Source: Macosko et al., 2015 *Cell*]

How does scRNA-seq technology work? Unique Molecular Identifiers



UMIs allow duplicate sequencing reads to be collapsed (de-duplicated), since each UMI must have originated from a single mRNA molecule

In other words, if two reads have the same Cell Barcode and UMI, they are a result of PCR duplication and can be collapsed to 1 “count” in the count matrix

[Source: Macosko et al., 2015 *Cell*]

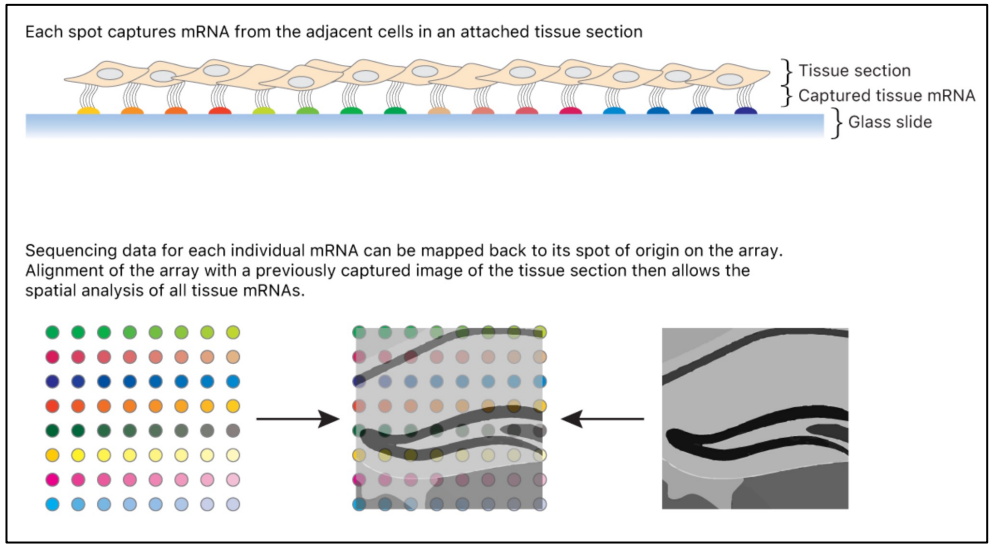
Single Cell Technologies: Variations & Further Reading

scATAC-seq (Assay for Transposase Accessible Chromatin)

- Measures chromatin accessibility in single nuclei
- Can be combined with scRNA-seq (in analysis step)
- [https://www.10xgenomics.com/products/single-cell-atac]

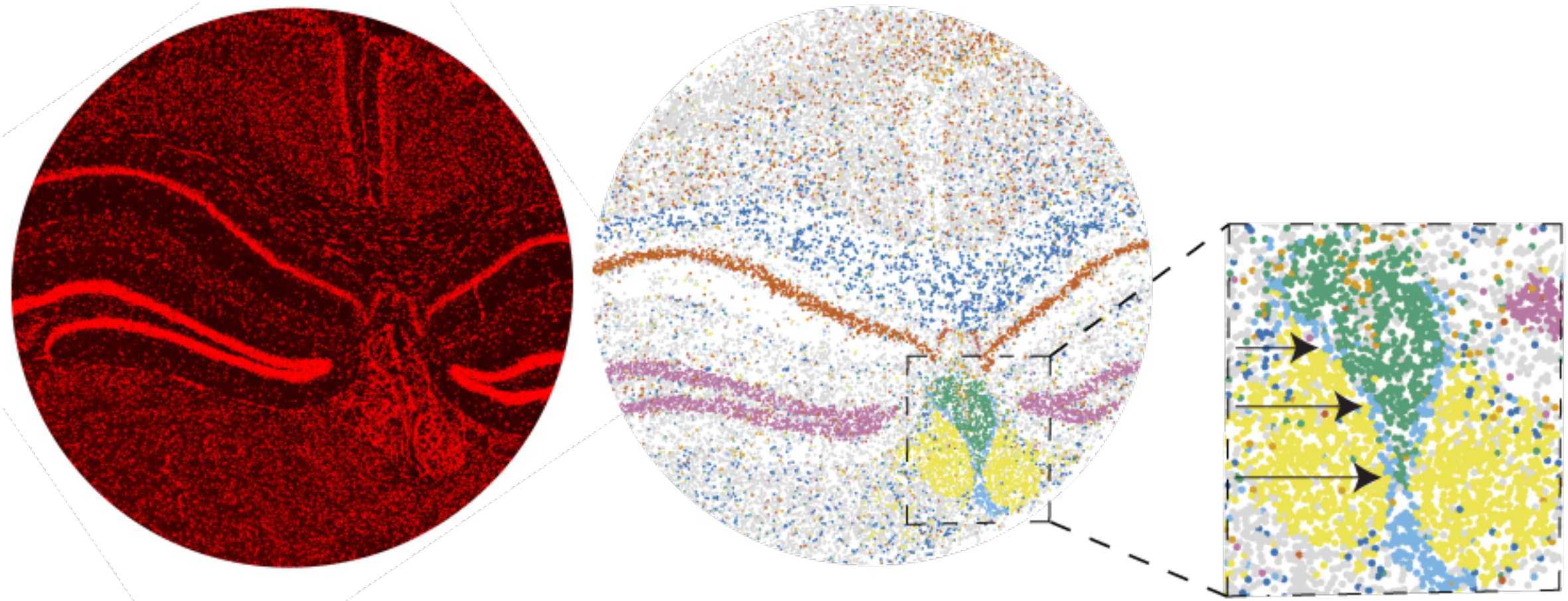
Spatial Transcriptomics

- Map single cell transcriptomes to spatial tissue location



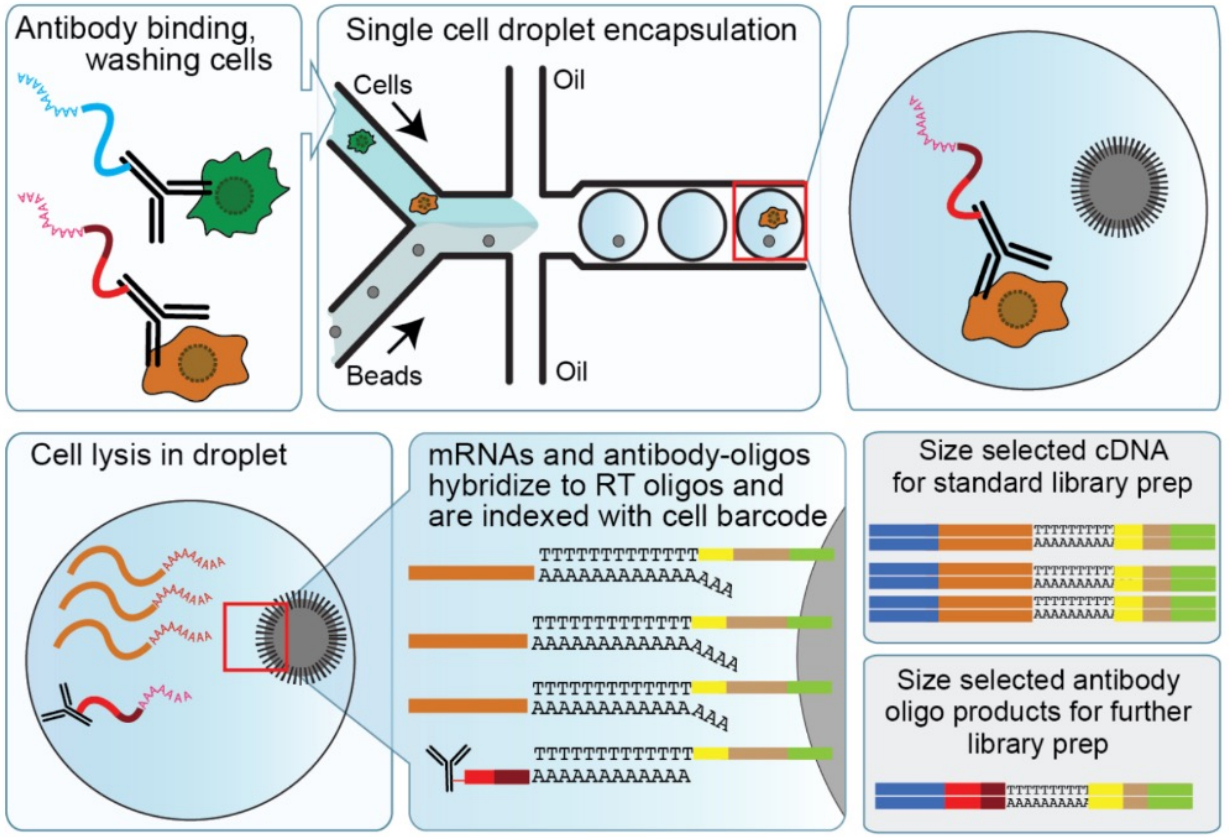
[Source: Wikipedia]

Single Cell Technologies: Variations & Further Reading



[Source: Broad Institute]

CITE-seq (Cellular Indexing of Transcriptomes and Epitopes by Sequencing)



[Source: https://www.protocols.io/view/cite-seq-and-cell-hashing-nhqdb5w.html]

Part II: Sample Preparation, Submission, QC, & Sequencing

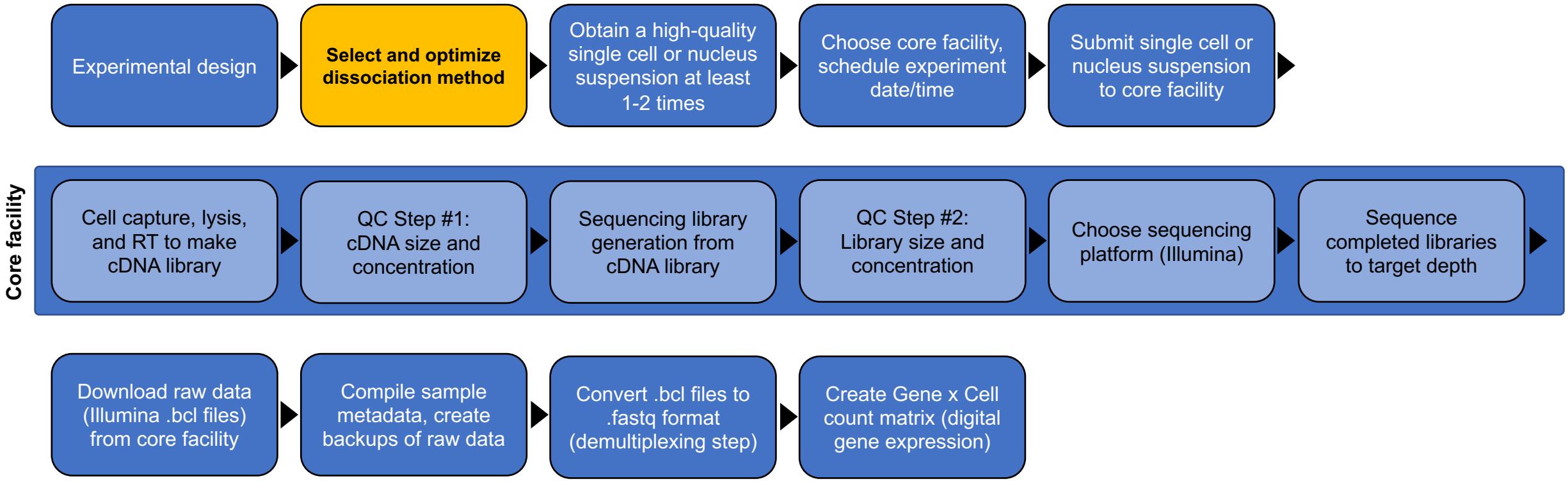
QC = “Quality Control”

Sample Preparation, Submission, QC, & Sequencing

- It is essential to consider ***and optimize*** sample preparation before running a single-cell experiment.
- In general, 10X Genomics products are plug-and-play ***if the input is high quality and free of all debris*** (both cellular and physical)
- Most failed experiments are caused by impure, low-quality, or poorly quantified input samples

Note: each lane/sample of 10X Genomics costs >$1000

Single Cell Experiment Outline



Tissue dissociation vs. nucleus extraction

Tissue Dissociation

The Good

- More RNA content in whole cells
- Ideal for less rigid or younger tissues
- Variety of established protocols
- Includes all cytoplasmic and nuclear RNA
- Compatible with CITE-seq

The Bad

- Requires long dissociation times
- Very challenging in adult brain tissue
- Large cells can clog microfluidic channels
- Harder to manage cellular debris
- Requires fresh tissue
- Different cell types in a tissue dissociate unevenly

Nucleus Extraction

The Good

- Compatible with most tissues/organisms without much protocol variation
- Compatible with frozen tissue
- Comparable discrimination of cell types as compared to single cell experiments[1]
- Rapid isolation time (< 60 min)
- Nuclei are more resistant to physical insult
- Compatible with scATAC-seq

The Bad

- Lower RNA content than whole cells
- Incompatible with CITE-seq*
- More ambient RNA in final sample

[1] Bakken et al., 2018, *Plos One*
* For cell surface antigens

Tissue Dissociation

Important Considerations

- Animal age
  - Ideally < P30 for mouse
- Tissue Rigidity
  - Not suited for rigid tissue
- Transcriptional inhibitors
  - Actinomysin D can be added to prevent stress-induced transcriptional responses
- Protease enzyme(s) choice
- Dissociation time
  - Must be optimized for each tissue/brain structure
- Dissociation conditions
  - Temperature, shaking/rocking speed, dissociation medium, tube angle, use foil instead of paper whenever possible

Reference Protocols

Saunders et al., 2018 *Cell* – Includes detailed descriptions of single cell isolation from multiple mouse brain structures

https://pubmed.ncbi.nlm.nih.gov/30096299/

Saxena et al., 2012 *Biotechniques* – Protocol for adult mouse brain dissociation (used in SLCR vmPFC isolation ~P30)

https://pubmed.ncbi.nlm.nih.gov/22668417/

Nucleus Extraction

Important Considerations

- Lysis method
  - Mechanical or detergent-based
- Ambient RNA levels
  - Generally higher than single cell experiments
- Dissociation conditions
  - Temperature (cold), use foil instead of paper whenever possible

Reference Protocol

Matson et al., 2018 JoVE – excellent protocol for rapid isolation of nuclei using mechanical or detergent-based lysis (protocol used in all SLCR nucleus extraction experiments). Compatible with fresh and frozen tissue.

https://pubmed.ncbi.nlm.nih.gov/30371670/

Available Protocols from 10X Genomics, Summer 2021

Library of 10X Genomics documented protocols:

https://support.10xgenomics.com/single-cell-gene-expression/sample-prep

Sample Prep

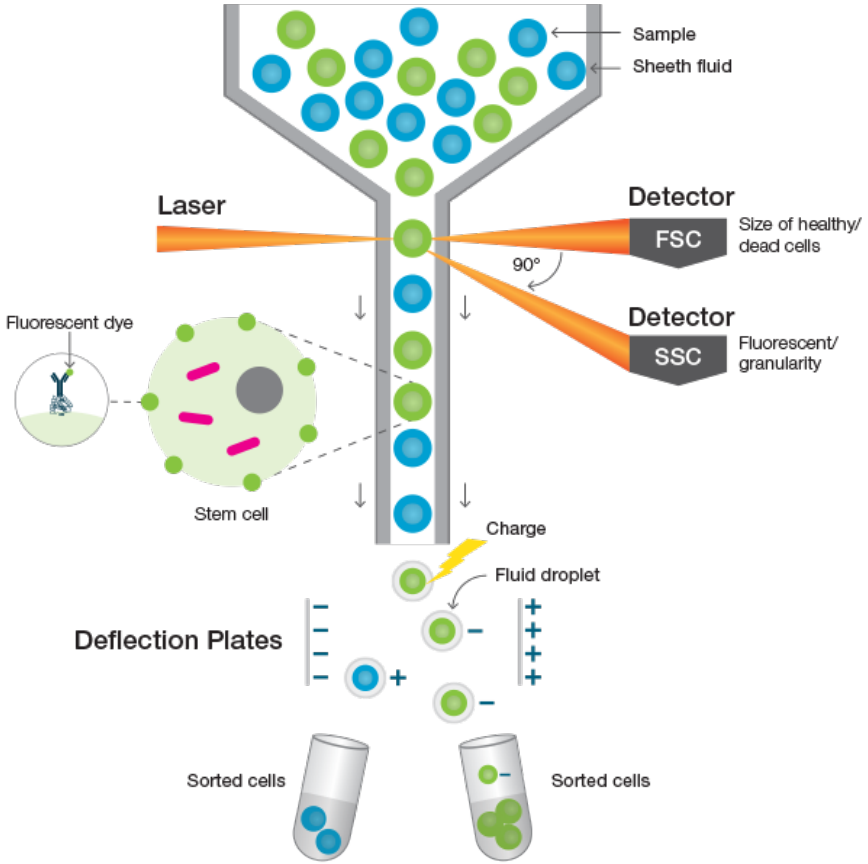
- Isolation of Leukocytes, Bone Marrow and Peripheral Blood Mononuclear Cells for Single Cell RNA Sequencing
- Single Cell Gene Expression Demonstrated Protocol Compatibility Table
- Single Cell Protocols - Cell Preparation Guide
- Cell Surface Protein Labeling for Single Cell RNA Sequencing Protocols
- Methanol Fixation of Cells for Single Cell RNA Sequencing
- Single Cell Suspensions from Cultured Cell Lines for Single Cell RNA Sequencing
- Isolation of Nuclei for Single Cell RNA Sequencing
- Enrichment of CD3+ T Cells from Dissociated Tissues for Single Cell RNA Sequencing and Immune Repertoire Profiling
- Thawing Dissociated Tumor Cells for Single Cell RNA Sequencing
- Tumor Dissociation for Single Cell RNA Sequencing
- Removal of Dead Cells from Single Cell Suspensions for Single Cell RNA Sequencing
- Nuclei Isolation from Adult Mouse Brain Tissue for Single Cell RNA Sequencing
- Moss Protoplast Suspension for Single Cell RNA Sequencing
- Fresh Frozen Human-Mouse Cell Line Mixtures for Single Cell RNA Sequencing
- Fresh Frozen Human Peripheral Blood Mononuclear Cells for Single Cell RNA Sequencing
- Dissociation of Mouse Embryonic Neural Tissue for Single Cell RNA Sequencing
- Cell Multiplexing Oligo Labeling for Single Cell RNA Sequencing Protocols

FACTS ON FACS (Fluorescence Activated Cell Sorting) **and other cell enrichment methods**

- FACS is commonly integrated into single cell workflows, including both droplet-based and plate-based experiments
- Can be used in combination nucleus or whole cell dissociation
- Generally requires a fluorescent marker of some kind (often genetically or virally introduced)

Common Use Cases

- Plate sorting for Smart-seq protocol
- When a pure, targeted cell population is desired
- If the cell type of interest is too rare to dissect (e.g. retinal ganglion cells) or extremely low abundance (e.g. ipRGCs)



[Source: BioRad]

Other cell isolation/enrichment methods

Manual cell picking

- Compatible with Smart-seq
- Not ideal for 10X Genomics

Single cell picking retina neurons:
https://pubmed.ncbi.nlm.nih.gov/22546911/

Patch-seq

- Compatible with Smart-seq
- Not ideal for 10X Genomics

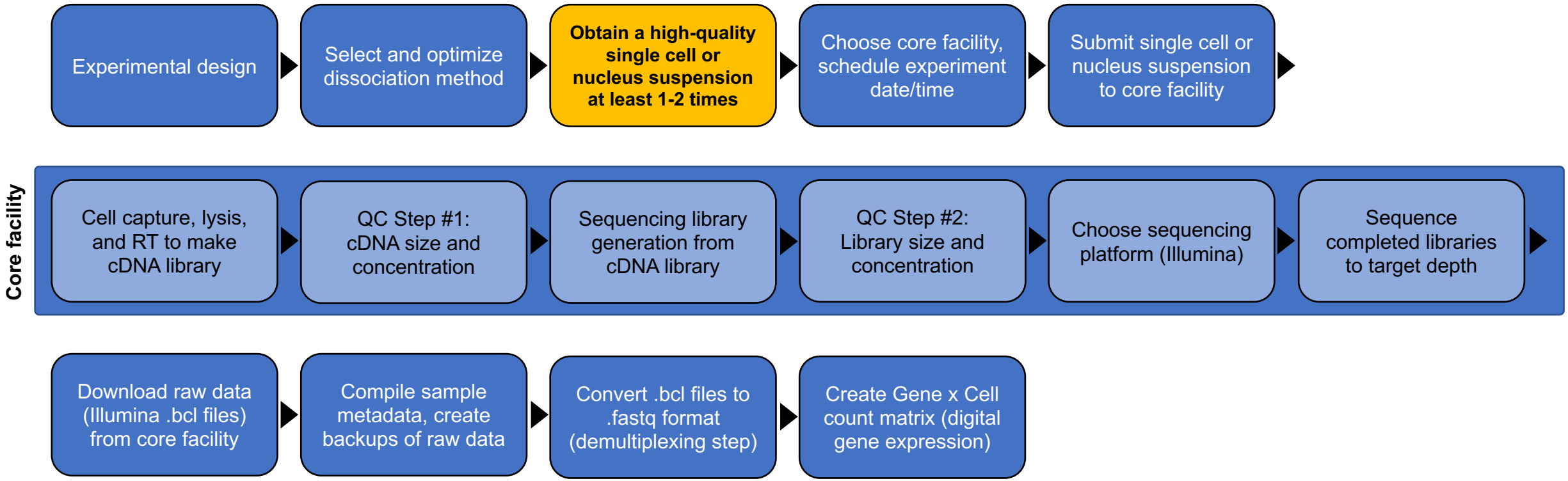
Original Patch-seq paper:
https://pubmed.ncbi.nlm.nih.gov/26689543/

Immunopanning

- Must be combined with another method for Smart-seq
- Compatible with 10X Genomics

Retinal ganglion cell immunopanning:
https://pubmed.ncbi.nlm.nih.gov/30018341/

Single Cell Experiment Outline



QC and assessing sample purity

Quality control is essential – samples *must* be clean and free of debris and/or cell clumps

Sample Handling Recommendations

- Avoid using paper-based products, instead use foil or plastic – fibers from Kimwipes and paper towels can *easily* clog microfluidics
- Consider using “lo-bind” tubes to prevent cells/nuclei/RNA from sticking to tube walls
- Use certified RNase-free reagents and consumables (tubes, etc.) and sterilize/RNaseZap all surfaces before collecting samples
- Carefully filter samples before QC and submission (35-100um filters, depending on tissue composition and cell isolation platform). Sometimes multiple filtration steps may be necessary.
- Whenever possible, keep samples on ice during processing

QC and assessing sample purity

Quality Control Measurements

Note: accurate sample concentration is critical for a successful experiment – avoid loading too many or too few cells

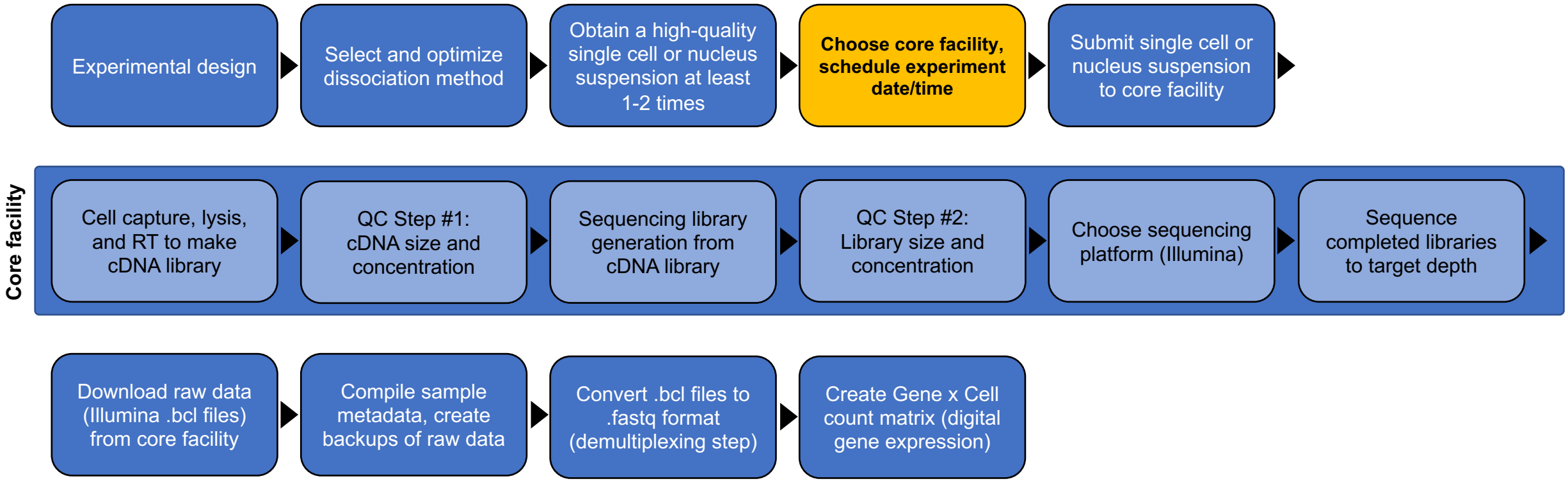
Whole cell dissociation

1. Measure cell concentration
   - Option 1: Sterile disposable hemocytometer with Trypan blue dye
   - Option 2: Automated cell counter with live/dead fluorescent dyes
2. Aim for at least ~80% live cells
3. Visually inspect the hemocytometer to ensure there are no large clumps or debris
   - If you see debris, filter the sample again and repeat hemocytometer measurement

Nucleus Extraction

1. Measure nucleus concentration
   - Option 1: Sterile disposable hemocytometer with 1:1 DAPI
   - Option 2: Automated cell counter (note: propidium iodide and other dead cell markers are designed to label nuclei – the counter will consider these as “dead” cells)
2. Visually inspect the hemocytometer to ensure there are no large clumps or debris
   - If you see debris, filter the sample again and repeat hemocytometer measurement

Single Cell Experiment Outline

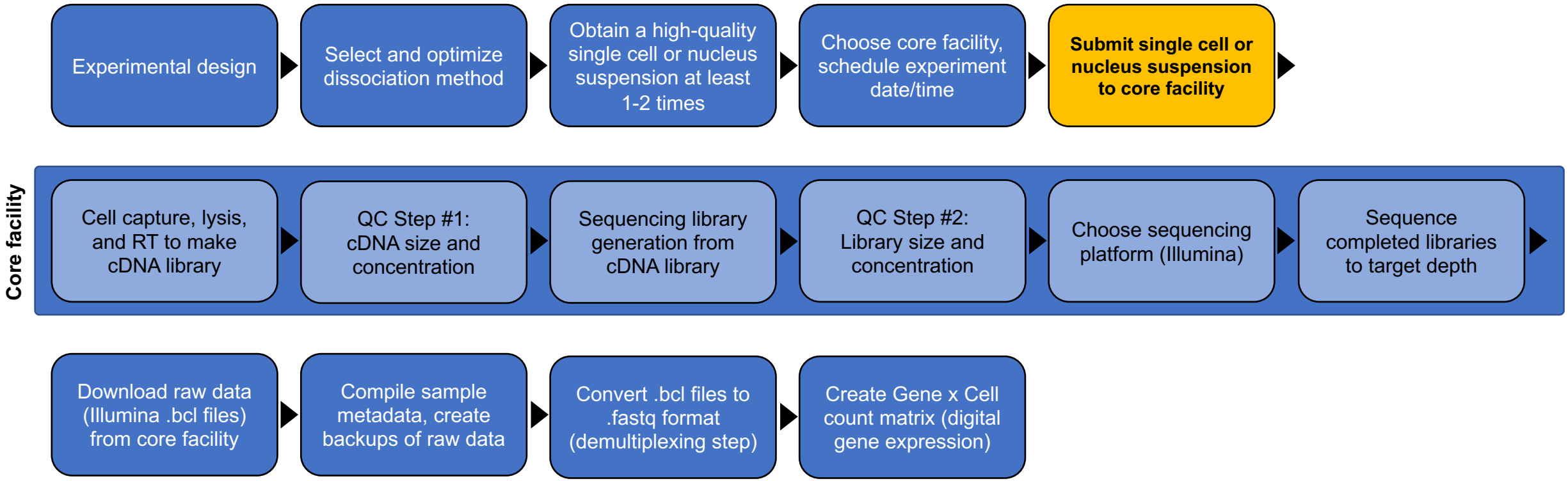


Sample Submission

Considerations when choosing a core facility

- Processing cost
- Does the core have previous experience?
  - With the platform
  - With similar model organisms
  - With similar tissues
  - Always good to ask for advice!
- Availability and location
  - Reduce transport time from lab → core as much as possible
- Equipment
  - What QC platforms do they have?
  - What sequencing platforms do they offer?
  - Will they quantify cell concentration in-house?
    - If so – which cell counter(s) are available?
- Turnaround and deliverables
  - How soon after submission will I receive QC?
  - How long is the queue for sequencing?
- Does the core offer any data analysis services?
- What reagents/consumables are supplied by the core facility?
- How can I download/access sequence data when my experiment is complete?
  - Recommended best practice: assume that all raw data is deleted from core facility servers after 1 week, so download, verify, and backup data within 1 week of completion

Single Cell Experiment Outline



SLCR Sample Submission – 10X Loading Concentrations

Step 1 GEM Generation & Barcoding

Cell Suspension Volume Calculator Table
(for step 1.2 of Chromium Next GEM Single Cell 3' v3.1 protocol)

Volume of Cell Suspension Stock per reaction (µl) | Volume of Nuclease-free Water per reaction (µl)

Cell Stock Concentration (Cells/µl)	Targeted Cell Recovery										
	500	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
100	8.3 35.0	16.5 26.7	33.0 10.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
200	4.1 39.1	8.3 35.0	16.5 26.7	24.8 18.5	33.0 10.2	41.3 2.0	n/a	n/a	n/a	n/a	n/a
300	2.8 40.5	5.5 37.7	11.0 32.2	16.5 26.7	22.0 21.2	27.5 15.7	33.0 10.2	38.5 4.7	n/a	n/a	n/a
400	2.1 41.1	4.1 39.1	8.3 35.0	12.4 30.8	16.5 26.7	20.6 22.6	24.8 18.5	28.9 14.3	33.0 10.2	37.1 6.1	41.3 2.0
500	1.7 41.6	3.3 39.9	6.6 36.6	9.9 33.3	13.2 30.0	16.5 26.7	19.8 23.4	23.1 20.1	26.4 16.8	29.7 13.5	33.0 10.2
600	1.4 41.8	2.8 40.5	5.5 37.7	8.3 35.0	11.0 32.2	13.8 29.5	16.5 26.7	19.3 24.0	22.0 21.2	24.8 18.5	27.5 15.7
700	1.2 42.0	2.4 40.8	4.7 38.5	7.1 36.1	9.4 33.8	11.8 31.4	14.1 29.1	16.5 26.7	18.9 24.3	21.2 22.0	23.6 19.6
800	1.0 42.2	2.1 41.1	4.1 39.1	6.2 37.0	8.3 35.0	10.3 32.9	12.4 30.8	14.4 28.8	16.5 26.7	18.6 24.6	20.6 22.6
900	0.9 42.3	1.8 41.4	3.7 39.5	5.5 37.7	7.3 35.9	9.2 34.0	11.0 32.2	12.8 30.4	14.7 28.5	16.5 26.7	18.3 24.9
1000	0.8 42.4	1.7 41.6	3.3 39.9	5.0 38.3	6.6 36.6	8.3 35.0	9.9 33.3	11.6 31.7	13.2 30.0	14.9 28.4	16.5 26.7
1100	0.8 42.5	1.5 41.7	3.0 40.2	4.5 38.7	6.0 37.2	7.5 35.7	9.0 34.2	10.5 32.7	12.0 31.2	13.5 29.7	15.0 28.2
1200	0.7 42.5	1.4 41.8	2.8 40.5	4.1 39.1	5.5 37.7	6.9 36.3	8.3 35.0	9.6 33.6	11.0 32.2	12.4 30.8	13.8 29.5
1300	0.6 42.6	1.3 41.9	2.5 40.7	3.8 39.4	5.1 38.1	6.3 36.9	7.6 35.6	8.9 34.3	10.2 33.0	11.4 31.8	12.7 30.5
1400	0.6 42.6	1.2 42.0	2.4 40.8	3.5 39.7	4.7 38.5	5.9 37.3	7.1 36.1	8.3 35.0	9.4 33.8	10.6 32.6	11.8 31.4
1500	0.6 42.7	1.1 42.1	2.2 41.0	3.3 39.9	4.4 38.8	5.5 37.7	6.6 36.6	7.7 35.5	8.8 34.4	9.9 33.3	11.0 32.2
1600	0.5 42.7	1.0 42.2	2.1 41.1	3.1 40.1	4.1 39.1	5.2 38.0	6.2 37.0	7.2 36.0	8.3 35.0	9.3 33.9	10.3 32.9
1700	0.5 42.7	1.0 42.2	1.9 41.3	2.9 40.3	3.9 39.3	4.9 38.3	5.8 37.4	6.8 36.4	7.8 35.4	8.7 34.5	9.7 33.5
1800	0.5 42.7	0.9 42.3	1.8 41.4	2.8 40.5	3.7 39.5	4.6 38.6	5.5 37.7	6.4 36.8	7.3 35.9	8.3 35.0	9.2 34.0
1900	0.4 42.8	0.9 42.3	1.7 41.5	2.6 40.6	3.5 39.7	4.3 38.9	5.2 38.0	6.1 37.1	6.9 36.3	7.8 35.4	8.7 34.5
2000	0.4 42.8	0.8 42.4	1.7 41.6	2.5 40.7	3.3 39.9	4.1 39.1	5.0 38.3	5.8 37.4	6.6 36.6	7.4 35.8	8.3 35.0

Grey boxes: Volumes that would exceed the allowable water volume in each reaction
Yellow boxes: Indicate a low transfer volume that may result in higher cell load variability
Blue boxes: Optimal range of cell stock concentration to maximize the likelihood of achieving the desired cell recovery target

Recommended Reading:

Complete 10X Chromium User Guide

https://support.10xgenomics.com/single-cell-gene-expression/library-prep/doc/user-guide-chromium-single-cell-3-reagent-kits-user-guide-v31-chemistry

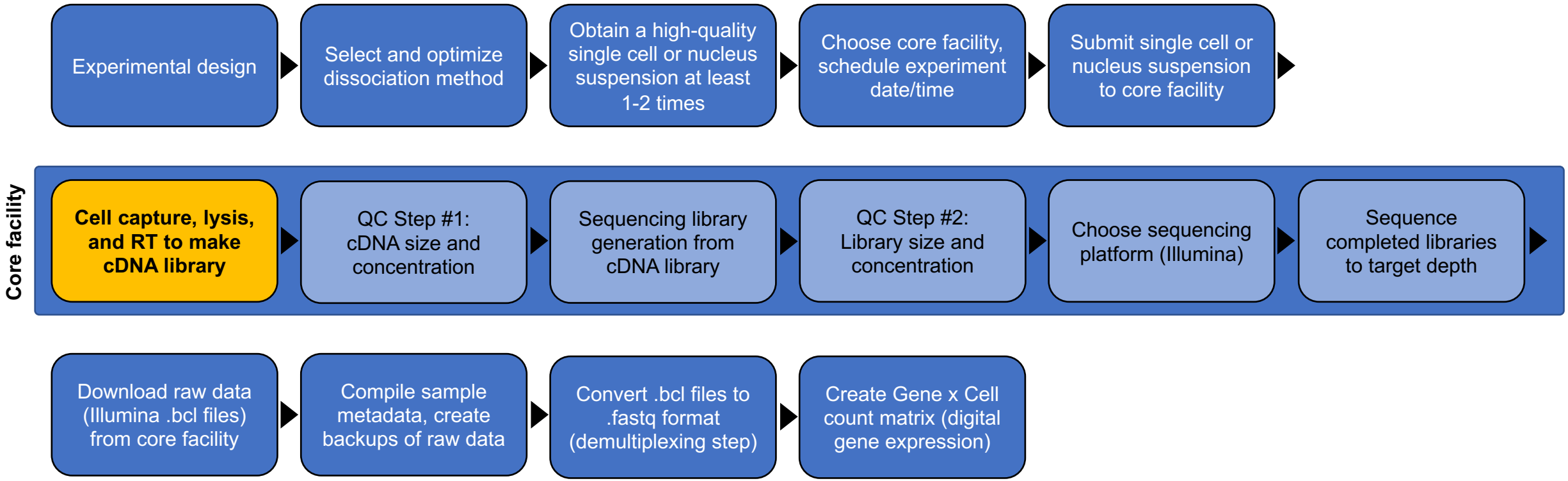
10X Genomics Cell Preparation Guide

https://assets.ctfassets.net/an68im79xiti/56DIUZEsVOWc8sSG42KQis/35cbcf6dcd4b0c0196263ee93815b0ae/CG000053_CellPrepGuide_RevC.pdf

(*includes 10X-compatible buffer list*)

[Source: 10X Genomics]

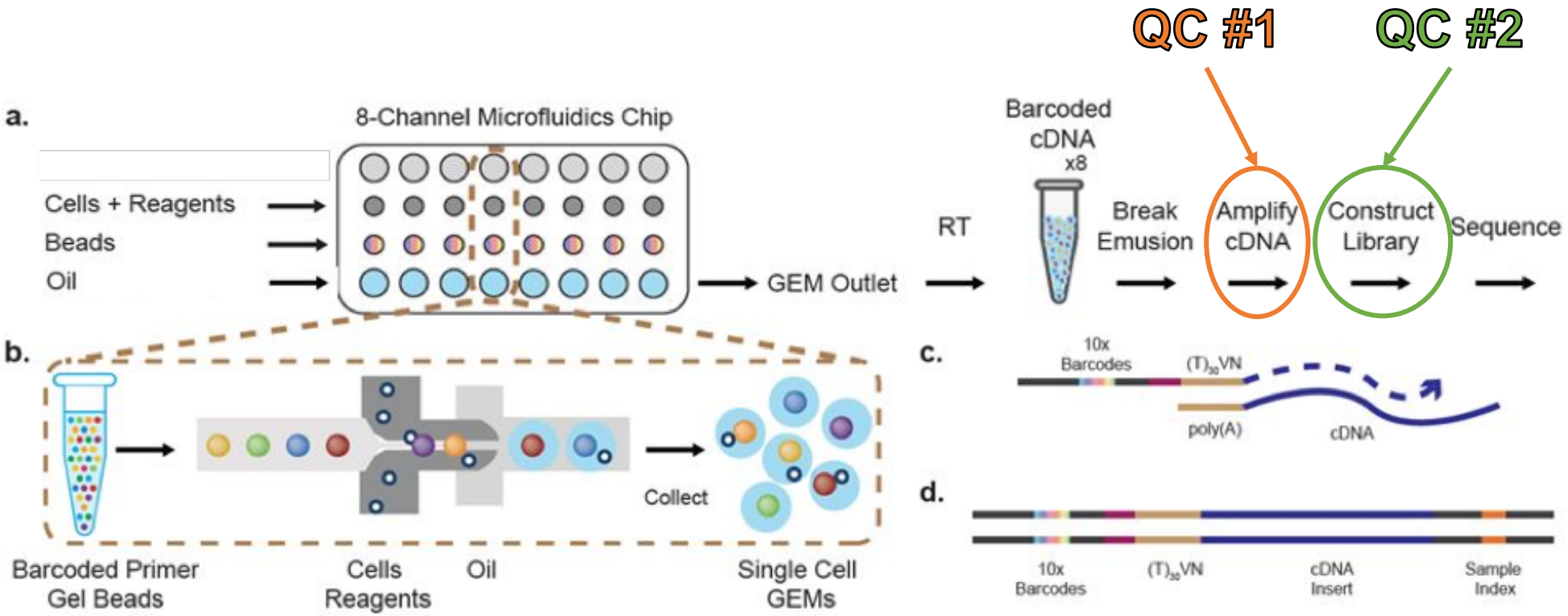
Single Cell Experiment Outline



10X Genomics Chromium Controller and Chip

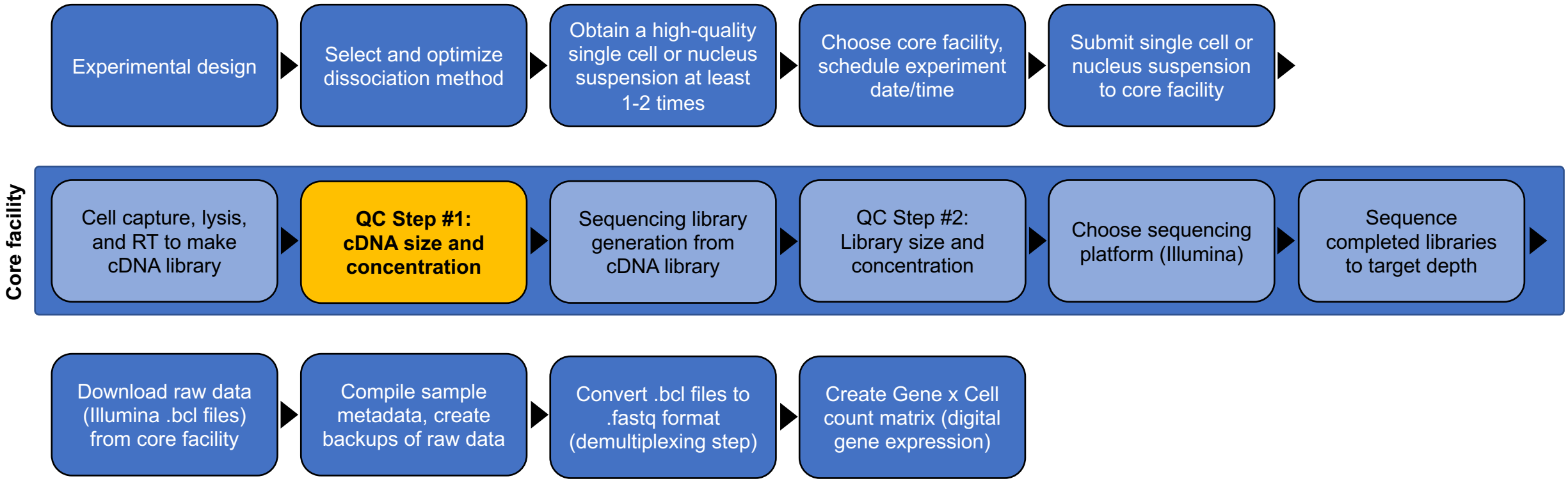
[Sources: medicine.uiowa.edu, 10X Genomics]

10X Genomics Chromium Controller – GEM generation and library construction



[Source: 10X Genomics]

Single Cell Experiment Outline



High-sensitivity assays to measure DNA size and concentration

Chip-based capillary electrophoresis

Agilent Bioanalyzer

- 11 Samples per chip
- Manual sample prep and chip loading

Agilent Tapestation

- Scalable to 96 samples per experiment
- Fully automated

[Source: Agilent Technologies]

High-sensitivity assays to measure DNA size and concentration

Services			
Instrument	**Chip or ScreenTape**	**Nucleic Acid Type**	**Concentration Range**
Agilent TapeStation	D1000	DNA	0.1 ng/ul to 50 ng/ul
Agilent TapeStation	Genomic	DNA	10 ng/ul to 100 ng/ul
Agilent TapeStation	High Sensitivity D1000	DNA	10 pg/ul to 1000 pg/ul
Agilent TapeStation	High Sensitivity D5000	DNA	10 pg/ul to 1000 pg/ul
Agilent TapeStation	High Sensitivity RNA	RNA	500 pg/ul to 10000 pg/ul
Agilent TapeStation	RNA	RNA	25 ng/ul to 500 ng/ul
Agilent BioAnalyzer	DNA High Sensitivity	DNA	5 pg/ul to 500 pg/ul
Agilent BioAnalyzer	RNA Nano	Total RNA	5 ng/ul to 500 ng/ul
Agilent BioAnalyzer	RNA 6000 Nano	mRNA	25 ng/ul to 250 ng/ul
Agilent BioAnalyzer	RNA Pico	Total RNA	50 pg/ul to 5000 pg/ul
Agilent BioAnalyzer	RNA 6000 Pico	mRNA	250 pg/ul to 5000 pg/ul
Agilent BioAnalyzer	Small RNA	small RNA*	50 pg/ul to 2000 pg/ul

* Size Range for small RNA is 6 to 150 nucleotides.

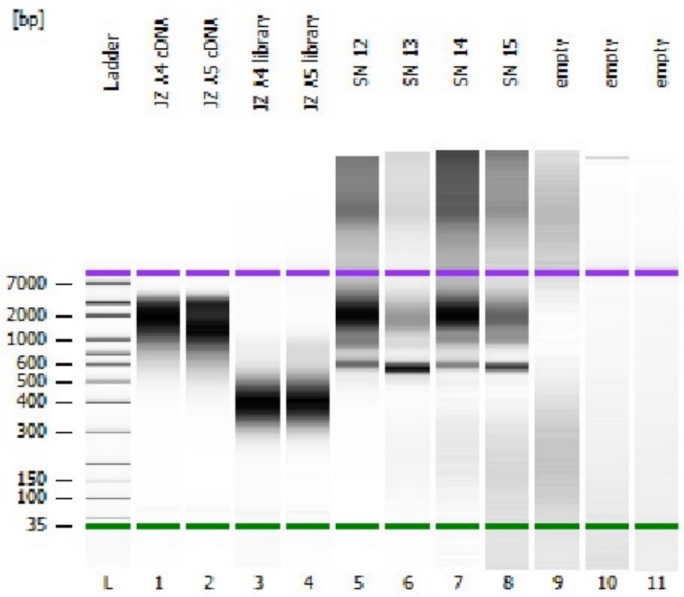
For additional information about the Bioanalyzer: https://www.agilent.com/cs/library/slidepresentation/Public/Overview%20Agilent%20Microfluidics.pdf

Technical performance comparison of Bioanalyzer vs. Tapestation: https://www.chem-agilent.com/pdf/5991-9093EN.PDF

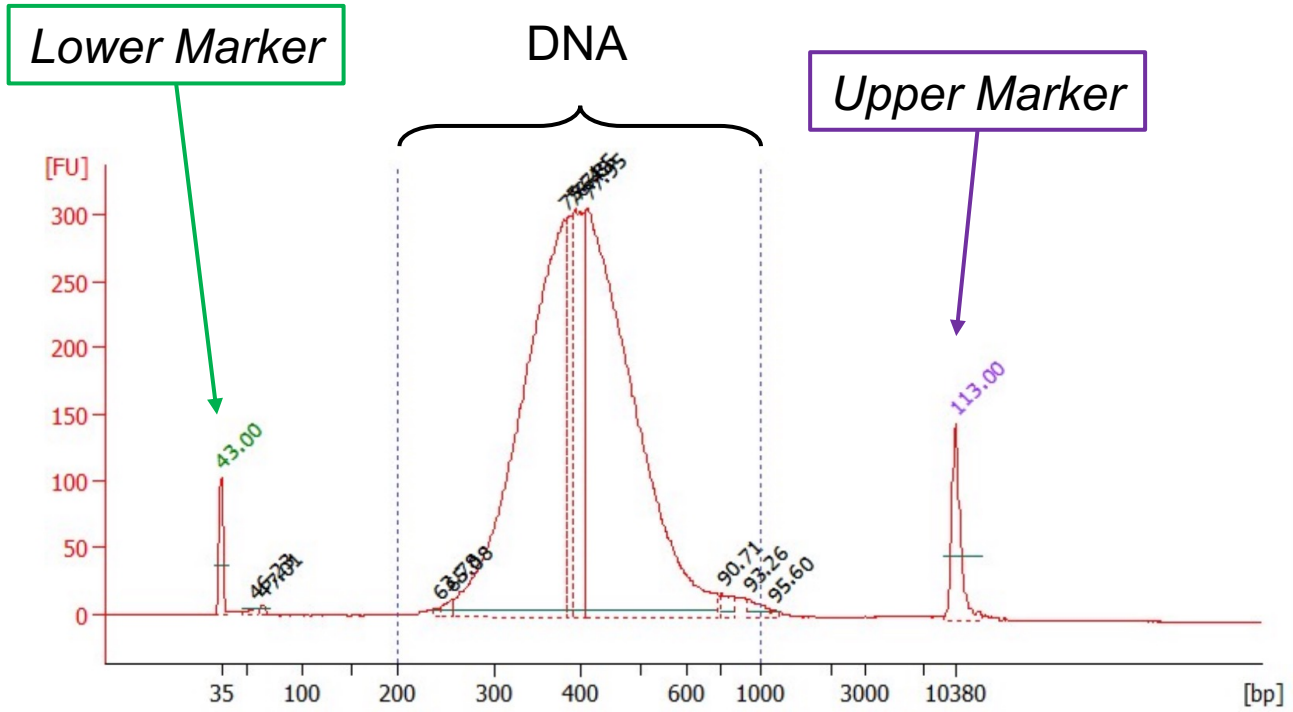
[Source: Harvard Medical School]

High-sensitivity assays to measure DNA size and concentration

Run Summary



Individual Sample Summary



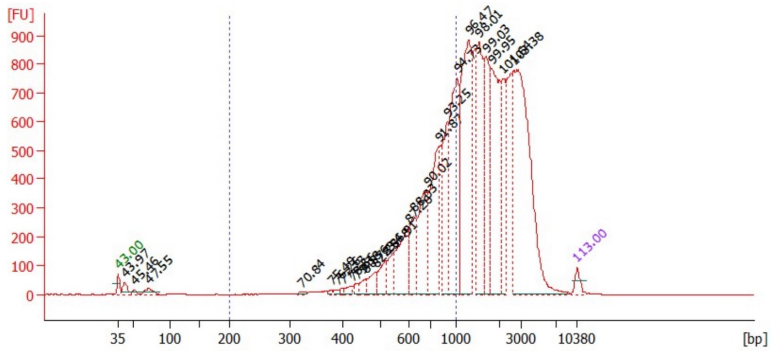
[Source: SLCR]

Single Cell RNA-seq cDNA Library QC

cDNA Size Concentration

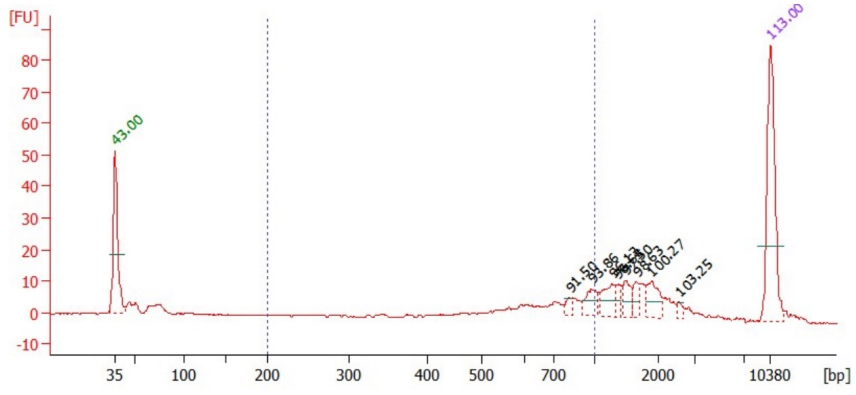
- Amplified via PCR based on targeted # of cells/nuclei
- Low concentration of cDNA could indicate:
  - Poor sample quality
  - Incorrect cell count
  - 10X microfluidic chip blockage

Good Trace



Good size, Good concentration

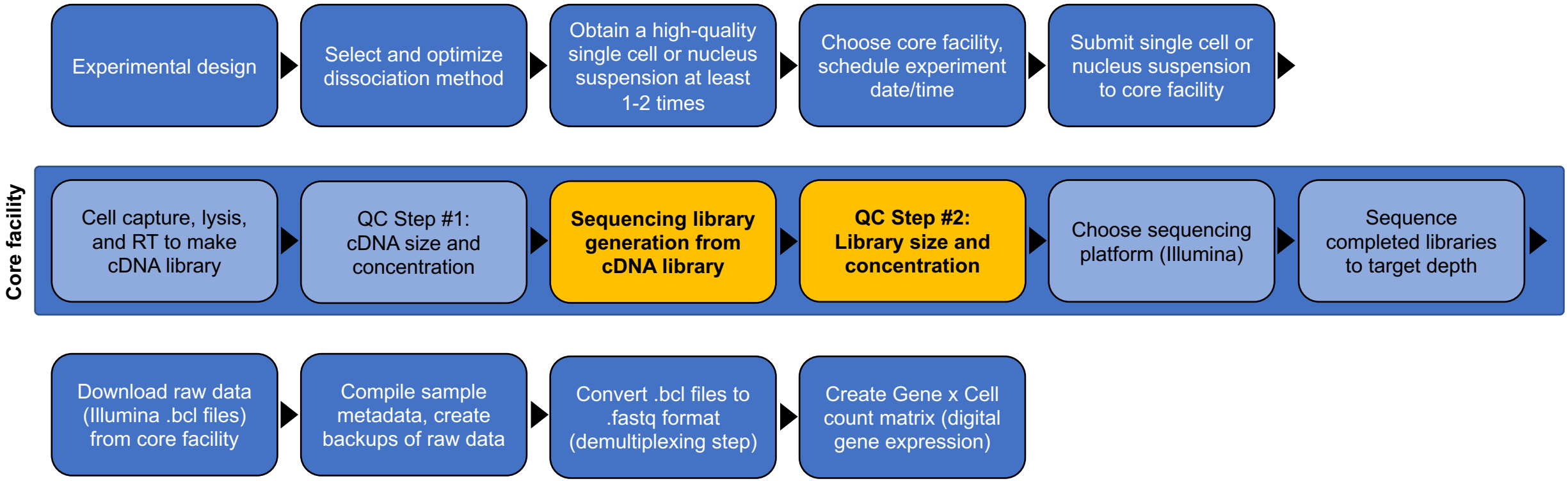
Not-so-good Trace



Good size, Poor concentration

[Source: SLCR]

Single Cell Experiment Outline

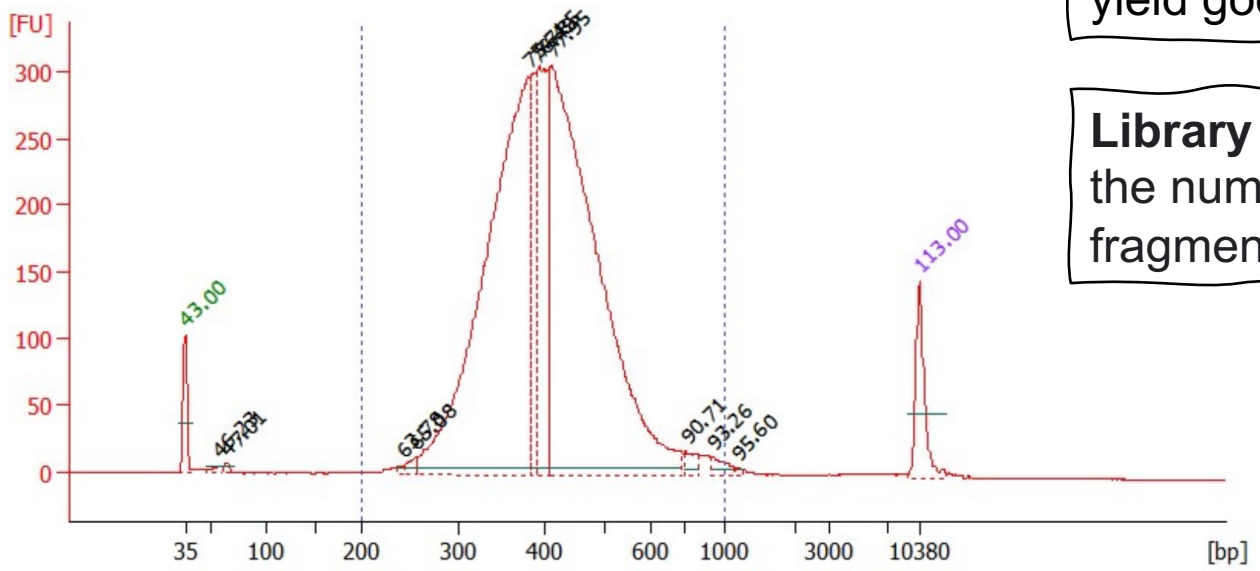


Single Cell RNA-seq Sequencing Library QC

Library Concentration

- Amplified via PCR based on targeted # of cells/nuclei
- Incorrect size could indicate:
  - Issue in enzymatic fragmentation step
    - Too small = incubation too long
    - Too large = incubation too short

Good Trace

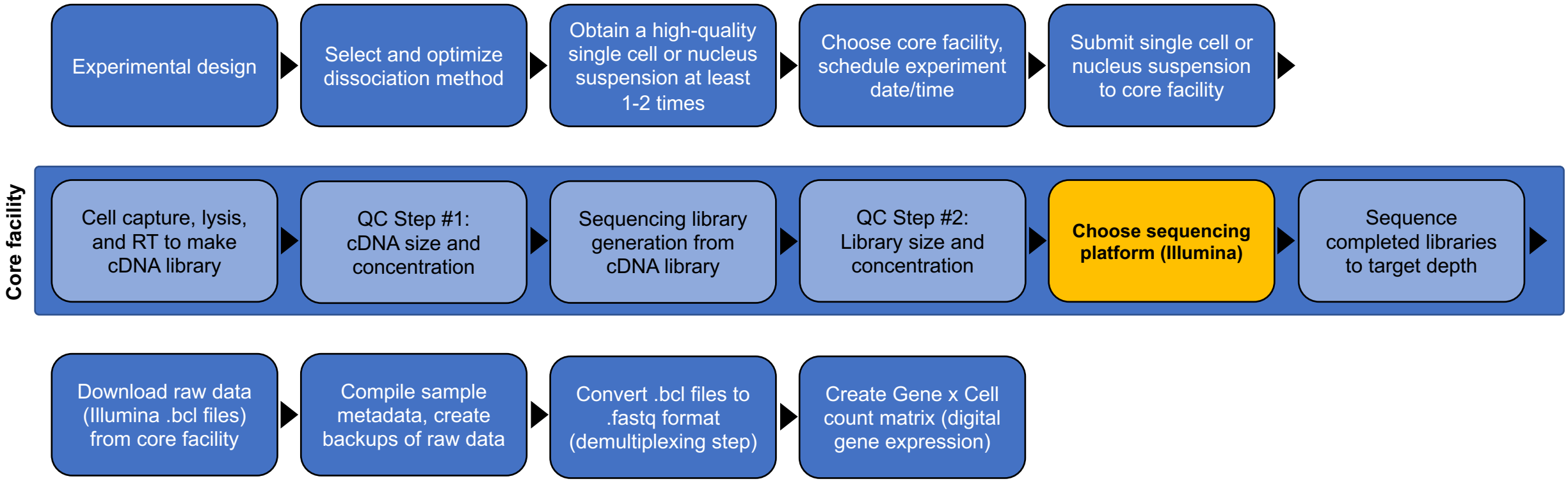


!! Note: poor cDNA concentration can result in libraries that "look" good based on size/concentration, but libraries generated from low quality cDNA will likely be low complexity and will not yield good data.

Library complexity refers to the number of *unique* DNA fragments present in **library**.

[Source: SLCR]

Single Cell Experiment Outline



Sequencing Technologies and Use Cases

Comparison of high-throughput sequencing methods[77][78]

Method	Read length	Accuracy (single read not consensus)	Reads per run	Time per run	Cost per 1 billion bases (in US$)	Advantages	Disadvantages
Single-molecule real-time sequencing (Pacific Biosciences)	30,000 bp (N50); maximum read length >100,000 bases[79][80][81]	87% raw-read accuracy[82]	4,000,000 per Sequel 2 SMRT cell, 100–200 gigabases[79][83][84]	30 minutes to 20 hours[79][85]	$7.2-$43.3	Fast. Detects 4mC, 5mC, 6mA.[86]	Moderate throughput. Equipment can be very expensive.
Ion semiconductor (Ion Torrent sequencing)	up to 600 bp[87]	99.6%[88]	up to 80 million	2 hours	$66.8-$950	Less expensive equipment. Fast.	Homopolymer errors.
Pyrosequencing (454)	700 bp	99.9%	1 million	24 hours	$10,000	Long read size. Fast.	Runs are expensive. Homopolymer errors.
Sequencing by synthesis (Illumina)	MiniSeq, NextSeq: 75–300 bp; MiSeq: 50–600 bp; HiSeq 2500: 50–500 bp; HiSeq 3/4000: 50–300 bp; HiSeq X: 300 bp	99.9% (Phred30)	MiniSeq/MiSeq: 1–25 Million; NextSeq: 130-00 Million; HiSeq 2500: 300 million – 2 billion; HiSeq 3/4000 2.5 billion; HiSeq X: 3 billion	1 to 11 days, depending upon sequencer and specified read length[89]	$5 to $150	Potential for high sequence yield, depending upon sequencer model and desired application.	Equipment can be very expensive. Requires high concentrations of DNA.
Combinatorial probe anchor synthesis (cPAS- BGI/MGI)	BGISEQ-50: 35-50bp; MGISEQ 200: 50-200bp; BGISEQ-500, MGISEQ-2000: 50-300bp[90]	99.9% (Phred30)	BGISEQ-50: 160M; MGISEQ 200: 300M; BGISEQ-500: 1300M per flow cell; MGISEQ-2000: 375M FCS flow cell, 1500M FCL flow cell per flow cell.	1 to 9 days depending on instrument, read length and number of flow cells run at a time.	$5– $120		
Sequencing by ligation (SOLiD sequencing)	50+35 or 50+50 bp	99.9%	1.2 to 1.4 billion	1 to 2 weeks	$60–130	Low cost per base.	Slower than other methods. Has issues sequencing palindromic sequences.[91]
Nanopore Sequencing	Dependent on library preparation, not the device, so user chooses read length (up to 2,272,580 bp reported[92]).	~92–97% single read	dependent on read length selected by user	data streamed in real time. Choose 1 min to 48 hrs	$7–100	Longest individual reads. Accessible user community. Portable (Palm sized).	Lower throughput than other machines, Single read accuracy in 90s.
GenapSys Sequencing	Around 150 bp single-end	99.9% (Phred30)	1 to 16 million	Around 24 hours	$667	Low-cost of instrument ($10,000)	
Chain termination (Sanger sequencing)	400 to 900 bp	99.9%	N/A	20 minutes to 3 hours	$2,400,000	Useful for many applications.	More expensive and impractical for larger sequencing projects. This method also requires the time-consuming step of plasmid cloning or PCR.

[Source: Wikipedia.org]

Illumina Sequencing by Synthesis

https://www.youtube.com/watch?v=fCd6B5HRaZ8

Illumina Sequencing by Synthesis

Some Important Illumina Sequencing Terms:

Index – the library identification barcode added during the sequencing library construction step. Indexes allow for multiple libraries to be sequenced in a single lane of Illumina sequencing. (10X Genomics indexes are numbered A1-A12, B1-B12, etc.)

Base call – the process of assigning a nucleotide base during a cycle of sequencing. Base calls include a quality score that denotes the confidence of the call.

Flow cell – a glass slide with multiple isolated "lanes" that is read by an Illumina sequencing machine. The flow cell allows for liquid samples to be flowed through in sequential sequencing steps.

Lane[1] – an isolated portion of an Illumina flow cell which contains millions of covalently attached sequence primer oligos which "catch" sequencing library fragments

Cluster[2] – a small "forest" of a clonally amplified sequencing library fragment attached to the lane of a flow cell. Each cluster contains identical DNA sequences, and each lane contains millions of clusters that are generated by clonal amplification of captured library fragments.

Cycle – a single round of Illumina sequencing (measuring all 4 base pairs). A typical 10X genomics experiment requires 127 "cycles" (28bp R1, 8bp Index, 91bp R2)

[1] *different from 10X Chromium "lane"*
[2] *different from scRNA-seq "cluster"*

Illumina Sequencing by Synthesis: Paired-End 10X Genomics Sequencing

'R1' – 28bp
10X Barcode
UMI

'R2' – 91bp
mRNA fragment

'I1' – 8bp
10X Library Index

A Survey of Illumina Sequencers

MiniSeq System

Power and simplicity for targeted sequencing.

MiSeq Series

Small genome and targeted sequencing.

NextSeq Series

Everyday genome, exome transcriptome sequencing, and more.

HiSeq Series

Production-scale genome, exome, transcriptome sequencing, and more.

NovaSeq Series

Population- and production-scale genome, exome, transcriptome sequencing, and more.

[Sources: https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf
https://www.illumina.com/systems/sequencing-platforms.html]

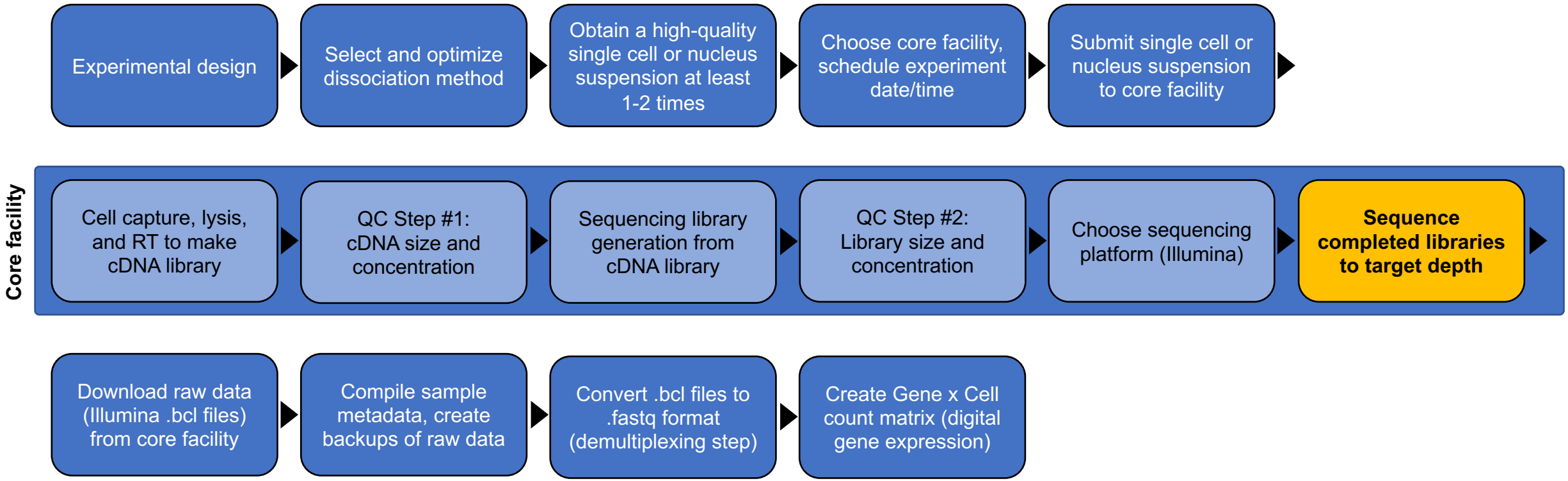
A Survey of Illumina Sequencers

	MiniSeq	MiSeq	NextSeq	HiSeq	NovaSeq
Max reads per run	25 million	25 million	400 million	2-3 billion	20 billion
Max read length	2 x 150bp	2 x 300bp	2 x 150bp	2 x 150bp	2 x 250bp
Lanes/flowcell	1	1	1*	2-8	2-4

[Sources: https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf
https://www.illumina.com/systems/sequencing-platforms.html
https://support.illumina.com/sequencing/sequencing_instruments/nextseq-500/questions.html]

*The NextSeq flow cell contains four physical lanes. However, libraries are loaded onto the flow cell from a single reservoir.

Single Cell Experiment Outline



10X Genomics Sequencing Recommendations

Single Cell 3' v3/v3.1 Gene Expression

The Chromium™ Single Cell Gene Expression Solution with Feature Barcode technology produces Illumina® sequencer-ready libraries.

Supported Sequencers

- Illumina® NovaSeq
- Illumina® HiSeq 3000/4000
- Illumina® HiSeq 2500 Rapid Run
- Illumina® NextSeq 500/550
- Illumina® MiSeq

Note: Sequencing libraries can be re-run to acquire additional reads and increase sequencing depth.

Recommended Sequencing: Minimum 20,000 read pairs/cell*

Dual Indexed Sequencing Run: Single Cell 3' v3/v3.1 libraries are single-indexed. We do not recommend sequencing 10x Single Cell 3' v3/v3.1 libraries with a dual-index configuration.

Read	Read 1	i7 Index	i5 Index	Read 2
Purpose	Cell barcode & UMI	Sample Index	N/A	Insert
Length**	28	8	0	91

[Source: https://support.10xgenomics.com/single-cell-gene-expression/sequencing/doc/specifications-sequencing-requirements-for-single-cell-3]

10X Genomics Sequencing Recommendations

NextSeq Series

NextSeq
400 million
2 x 150bp
1*

Example: running samples on an Illumina NextSeq

$$\frac{400\ million\ read\ pairs/run}{\left(20{,}000\ \frac{read\ pairs}{cell}\right)} = 20{,}000\ cells/run$$

Target = ~7k-8k cells/library

So, multiplex **2-3 libraries/run**

[Source: https://support.10xgenomics.com/single-cell-gene-expression/sequencing/doc/specifications-sequencing-requirements-for-single-cell-3]

Break!

Windows users, please install https://gitforwindows.org/

Part III: Workshop - Using the Linux/Unix command line

Shell (computing)

From Wikipedia, the free encyclopedia

In computing, a **shell** is a computer program which exposes an operating system's services to a human user or other programs. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. It is named a shell because it is the outermost layer around the operating system.[1][2]

Unix shell

From Wikipedia, the free encyclopedia

A **Unix shell** is a command-line interpreter or shell that provides a command line user interface for Unix-like operating systems. The shell is both an interactive command language and a scripting language, and is used by the operating system to control the execution of the system using shell scripts.[2]

[Source: Wikipedia.org]

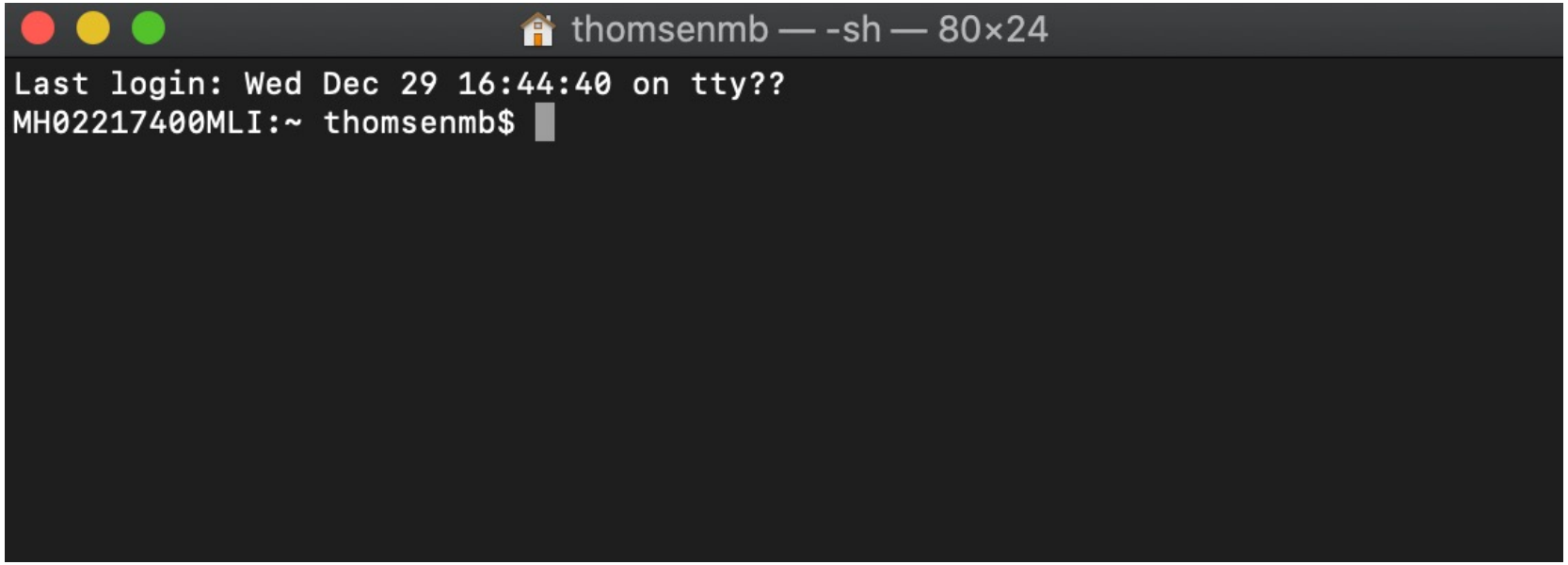
Bash (Unix shell)

From Wikipedia, the free encyclopedia

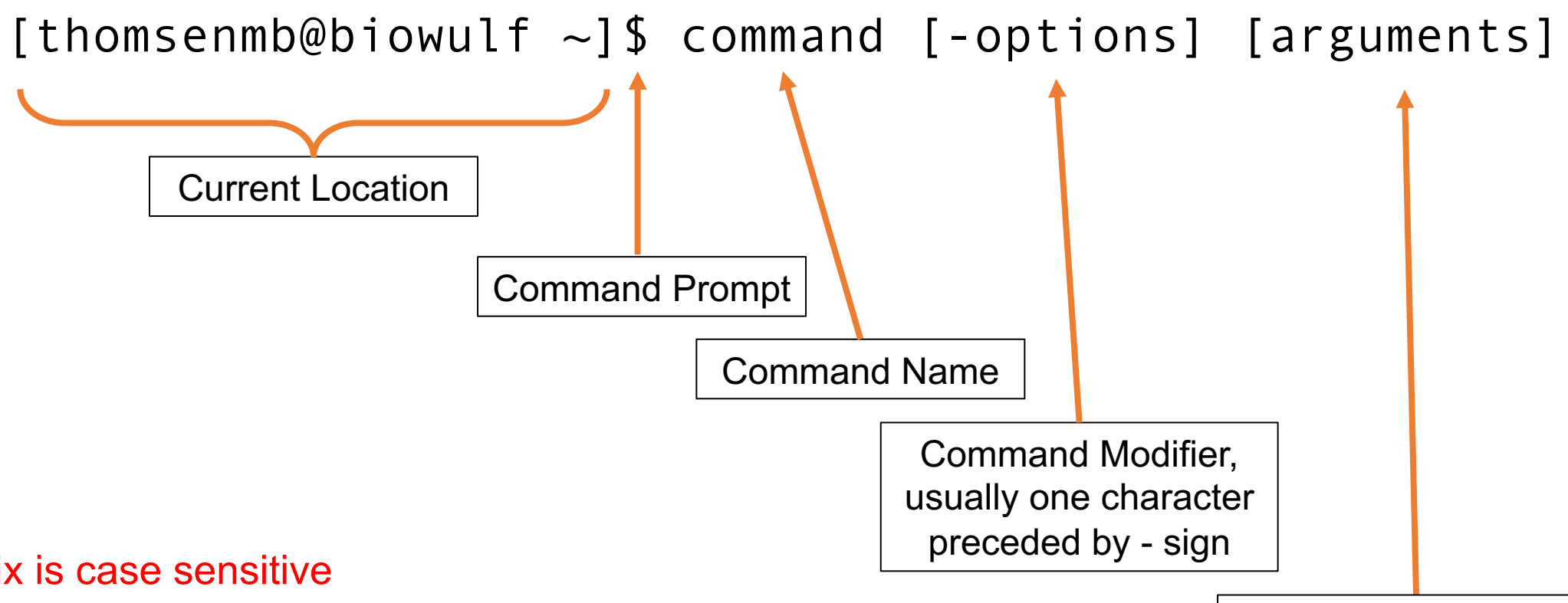
Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell.[10][11] First released in 1989,[12] it has been used as the default login shell for most Linux distributions.[13] A version is also available for Windows 10 via the Windows Subsystem for Linux.[14] It is also the default user shell in Solaris 11.[15] Bash was also the default shell in all versions of Apple macOS prior to the 2019 release of macOS Catalina, which changed the default shell to zsh, although Bash remains available as an alternative shell.[16]

[Source: Wikipedia.org]

Intro to the Command Line Interface (CLI)



Linux Command Line Reference



- Unix is case sensitive
- Fields enclosed in [] are not always required
- No space between minus sign and option modifier
- Leave spaces between command, options, and arguments

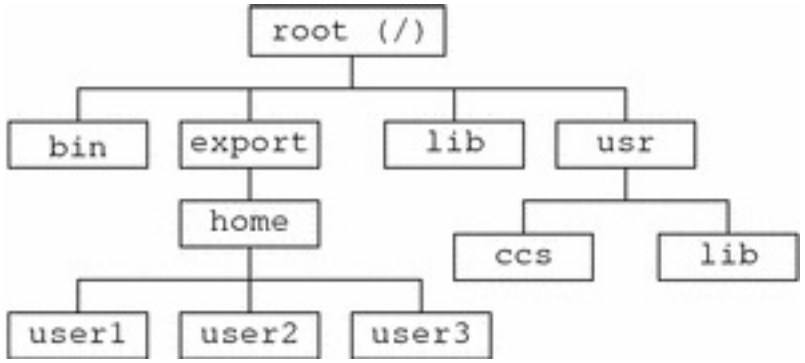
Linux Command Line Reference

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1* *file2* – copy *file1* to *file2*
cp -r *dir1* *dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1* *file2* – rename or move *file1* to *file2*
if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file* *link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

[Source: https://files.fosswire.com/2007/08/fwunixref.pdf
https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf]

Linux Full vs. Relative Paths



Absolute Path: the 'complete' path of a file or directory relative to the root (/) directory

- Always begins with '/'

Relative Path: the path of a file or directory relative to the current location (working directory)

Linux Command Line Reference

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

USE WITH EXTREME CAUTION

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.

|	Pipe (redirect) output
sudo [command]	run < command> in superuser mode
nohup [command]	**run < command> immune to hangup signal**
man [command]	display help pages of < command>
[command] &	**run < command> and send task to background**
>> [fileA]	append to fileA, preserving existing contents
> [fileA]	**output to fileA, overwriting contents**
echo -n	display a line of text

[Source: https://files.fosswire.com/2007/08/fwunixref.pdf https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf]

Linux Command Line Reference: File archive and compression

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

USE WITH EXTREME CAUTION

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Shortcuts

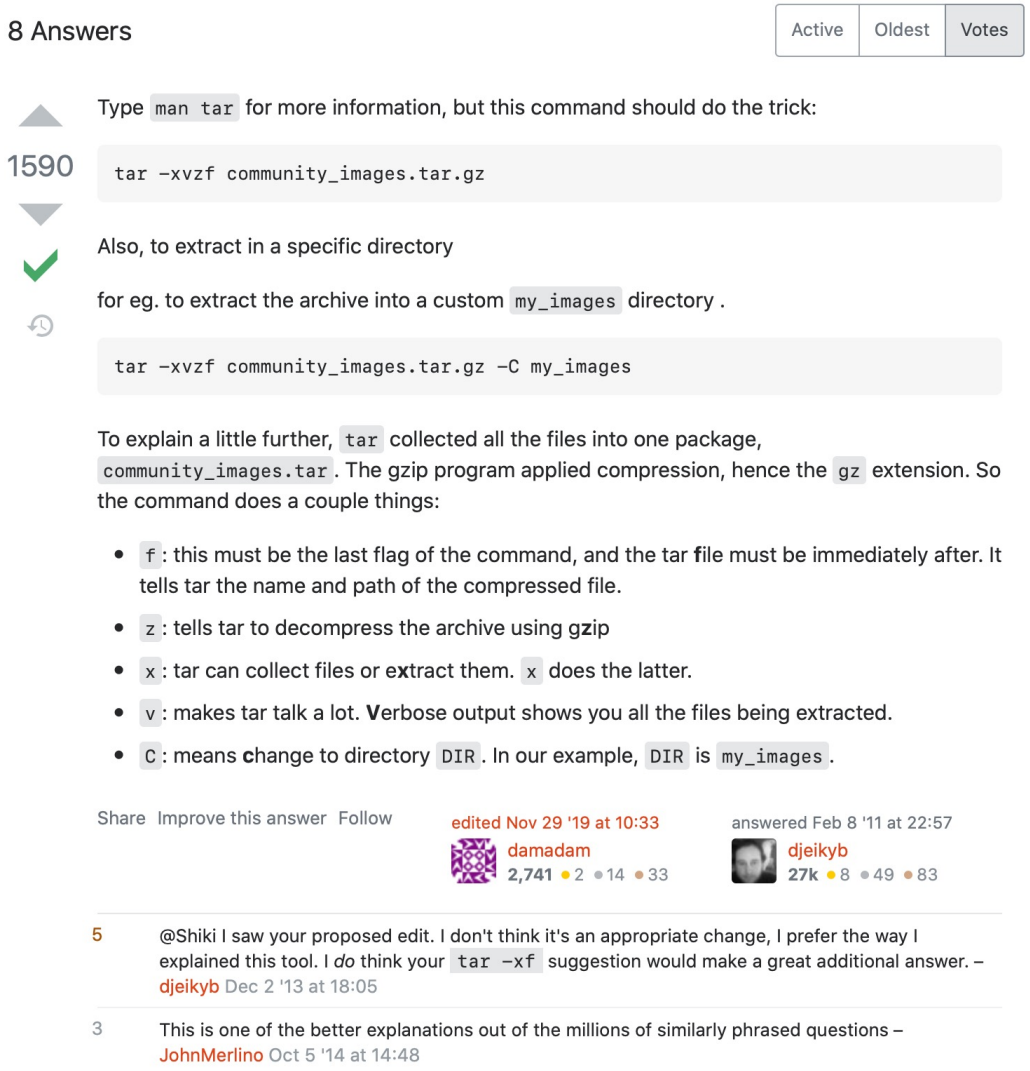
Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.

|	Pipe (redirect) output
sudo [command]	run < command> in superuser mode
nohup [command]	**run < command> immune to hangup signal**
man [command]	display help pages of < command>
[command] &	**run < command> and send task to background**
>> [fileA]	append to fileA, preserving existing contents
> [fileA]	**output to fileA, overwriting contents**
echo -n	display a line of text

[Source: https://files.fosswire.com/2007/08/fwunixref.pdf https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf]

Linux Command Line Reference: File archive and compression



8 Answers

Active | Oldest | Votes

1590

Type `man tar` for more information, but this command should do the trick:

```
tar -xvzf community_images.tar.gz
```

Also, to extract in a specific directory

for eg. to extract the archive into a custom `my_images` directory .

```
tar -xvzf community_images.tar.gz -C my_images
```

To explain a little further, `tar` collected all the files into one package, `community_images.tar`. The gzip program applied compression, hence the `gz` extension. So the command does a couple things:

- `f`: this must be the last flag of the command, and the tar **f**ile must be immediately after. It tells tar the name and path of the compressed file.
- `z`: tells tar to decompress the archive using g**z**ip
- `x`: tar can collect files or e**x**tract them. `x` does the latter.
- `v`: makes tar talk a lot. **V**erbose output shows you all the files being extracted.
- `C`: means **c**hange to directory `DIR`. In our example, `DIR` is `my_images`.

Share Improve this answer Follow

edited Nov 29 '19 at 10:33 damadam 2,741 ● 2 ● 14 ● 33

answered Feb 8 '11 at 22:57 djeikyb 27k ● 8 ● 49 ● 83

5 @Shiki I saw your proposed edit. I don't think it's an appropriate change, I prefer the way I explained this tool. I *do* think your `tar -xf` suggestion would make a great additional answer. – djeikyb Dec 2 '13 at 18:05

3 This is one of the better explanations out of the millions of similarly phrased questions – JohnMerlino Oct 5 '14 at 14:48

A great example of a StackExchange Q&A

https://askubuntu.com/questions/25347/what-command-do-i-need-to-unzip-extract-a-tar-gz-file

Using the Linux Command Line

```
Last login: Wed Jun  2 17:10:30 2021 from 10.243.231.36
[[thomsenmb@biowulf ~]$ cd /data/thomsenmb
[[thomsenmb@biowulf thomsenmb]$ ls -l
total 20
drwxr-x---.  4 thomsenmb thomsenmb 4096 May 27 18:02 bs-downloads
drwxr-xr-x.  4 thomsenmb thomsenmb 4096 Jun  1 20:37 cellranger-processed-data
drwxr-xr-x. 14 thomsenmb thomsenmb 8192 Jun  3 01:09 illumina-raw-data
drwxr-x---.  2 thomsenmb thomsenmb 4096 May 27 18:02 illumina-raw-data-tarballs
drwxr-x---.  5 thomsenmb thomsenmb 4096 May 27 18:17 marcc-fastq-files
[[thomsenmb@biowulf thomsenmb]$ cd illumina-raw-data-tarballs/
[[thomsenmb@biowulf illumina-raw-data-tarballs]$ ls -l
total 224523905
-rw-r-----. 1 thomsenmb thomsenmb 28535059936 May 27 16:15 20200123_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 26987177532 May 27 16:39 20200213_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 26185789677 May 27 17:02 20200723_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 23488541085 May 21 14:27 20201105_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 24923308684 May 27 17:24 20201109_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 32709309868 May 27 17:53 20201221_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 32914588751 May 19 14:40 20210324_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb 34168234332 May 19 13:59 20210412_nextseq_run.tar.gz
-rw-r-----. 1 thomsenmb thomsenmb         920 May 27 18:01 readme.txt
[[thomsenmb@biowulf illumina-raw-data-tarballs]$ cd ../cellranger-processed-data/
[[thomsenmb@biowulf cellranger-processed-data]$ ls
01232020_processed  03242021_processed
[[thomsenmb@biowulf cellranger-processed-data]$ cd 03242021_processed/demultiplexed/H3MGYBGXG/
[[thomsenmb@biowulf H3MGYBGXG]$ ls
MT39  MT39_count.sh  MT39_out  MT40  MT40_count.sh  MT40_out  MT41  MT41_count.sh  MT41_out  slurm-16343211.out  slurm-16374688.out  slurm-16374882.out
[[thomsenmb@biowulf H3MGYBGXG]$ less slurm-16343211.out
[[thomsenmb@biowulf H3MGYBGXG]$ less MT39_count.sh
```

Using the Linux Command Line

Interactive Portion

- Navigate to your ~ (home) directory
- Make a new directory called “sc-workshop-day1”
- Navigate to that directory
- Output the current directory
- Create a text file named whereami.txt containing the name of the current directory
- Use “less” to view the contents of whereami.txt
- List the contents of the current directory (with details)
- Move whereami.txt to your home directory
- Add a new line to whereami.txt that reads “end of file”
- Use “more” to check that your addition was successful
- Read the manual page for “more”
- Read the manual page for “less”
- Make a copy of whereami.txt in your sc-workshop-day1 directory
- Rename whereami.txt to “newname.txt”
- Delete newname.txt
- Delete whereami.txt

Be careful when copy pasting code snippets – Microsoft and other programs may not have Unix-compatible character formats, (e.g. - vs –)

Using the Linux Command Line

Interactive Portion

- Navigate to your ~ (home) directory
- Make a new directory called “sc-workshop-day1”
- Navigate to that directory
- Output the current directory
- Create a text file named whereami.txt containing the name of the current directory
- Use “less” to view the contents of whereami.txt
- List the contents of the current directory (with details)
- Move whereami.txt to your home directory
- Add a new line to whereami.txt that reads “end of file”
- Use “more” to check that your addition was successful
- Read the manual page for “more”
- Read the manual page for “less”
- Make a copy of whereami.txt in your sc-workshop-day1 directory
- Rename whereami.txt to “newname.txt”
- Delete newname.txt
- Delete whereami.txt

Be careful when copy pasting code snippets – Microsoft and other programs may not have Unix-compatible character formats, (e.g. - vs –)

“Answer Key”

- cd ~
- mkdir sc-workshop-day1
- cd sc-workshop-day1
- pwd
- pwd > whereami.txt
- less whereami.txt
- ls -l
- mv whereami.txt ../
- echo “end of file” >> whereami.txt
- more whereami.txt
- man more
- man less
- cp whereami.txt sc-workshop-day1
- mv whereami.txt newname.txt
- rm newname.txt
- rm sc-workshop-day1/whereami.txt

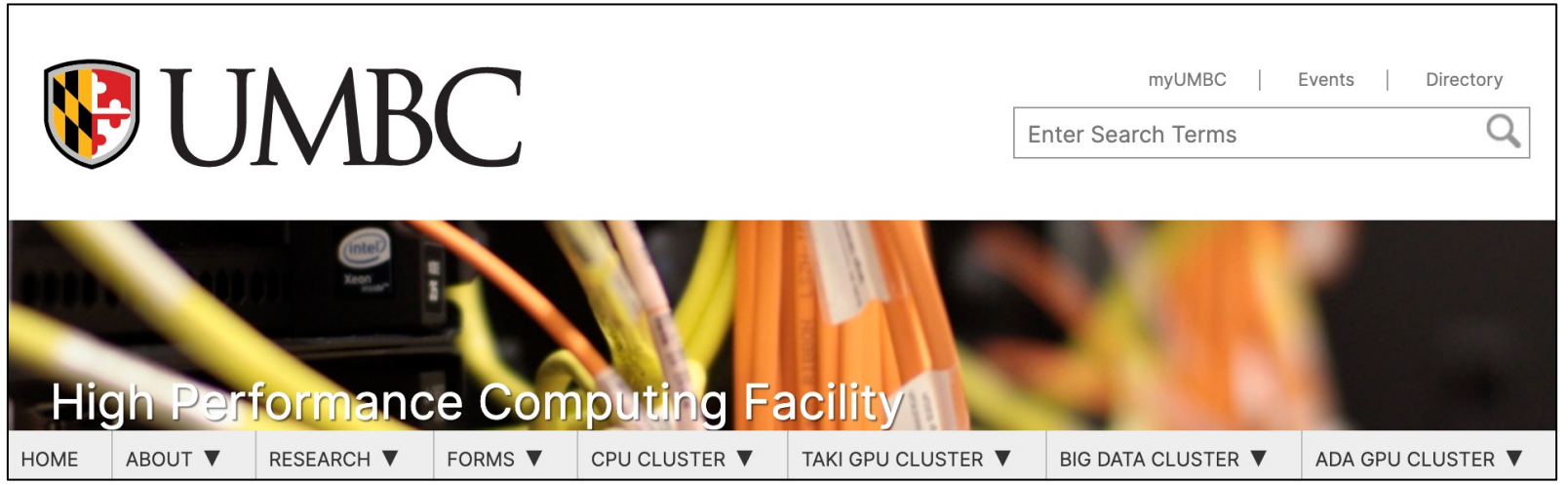
Break!

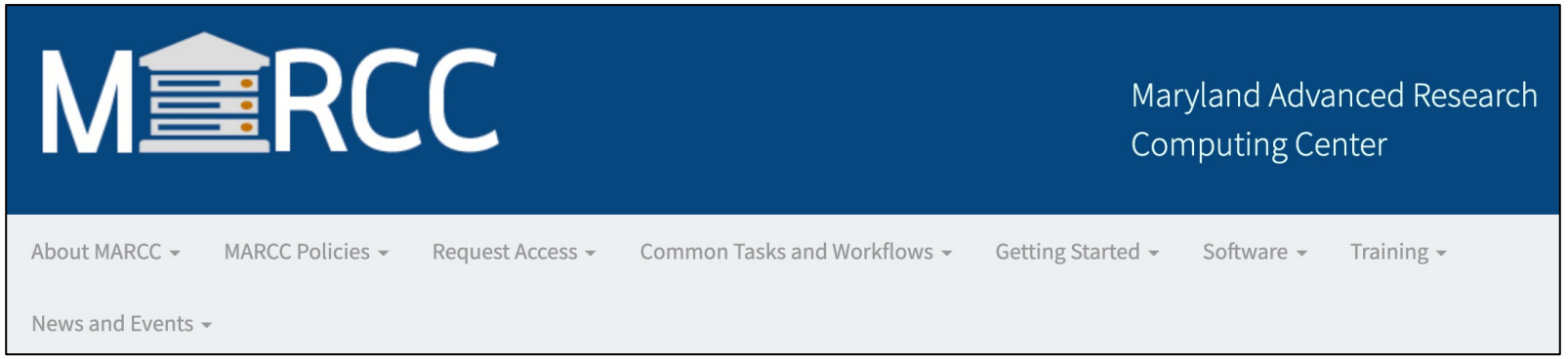
Part IV: Data Analysis: Downloading and Preprocessing Raw Data

Goal: transform raw sequencing data into a “count matrix”

Cell:	1	2	···	*N*
GENE 1	1	2		14
GENE 2	4	27		8
GENE 3	0	0		1
⋮	⋮	⋮		⋮
GENE M	6	2		0

Introduction to High Performance Computing



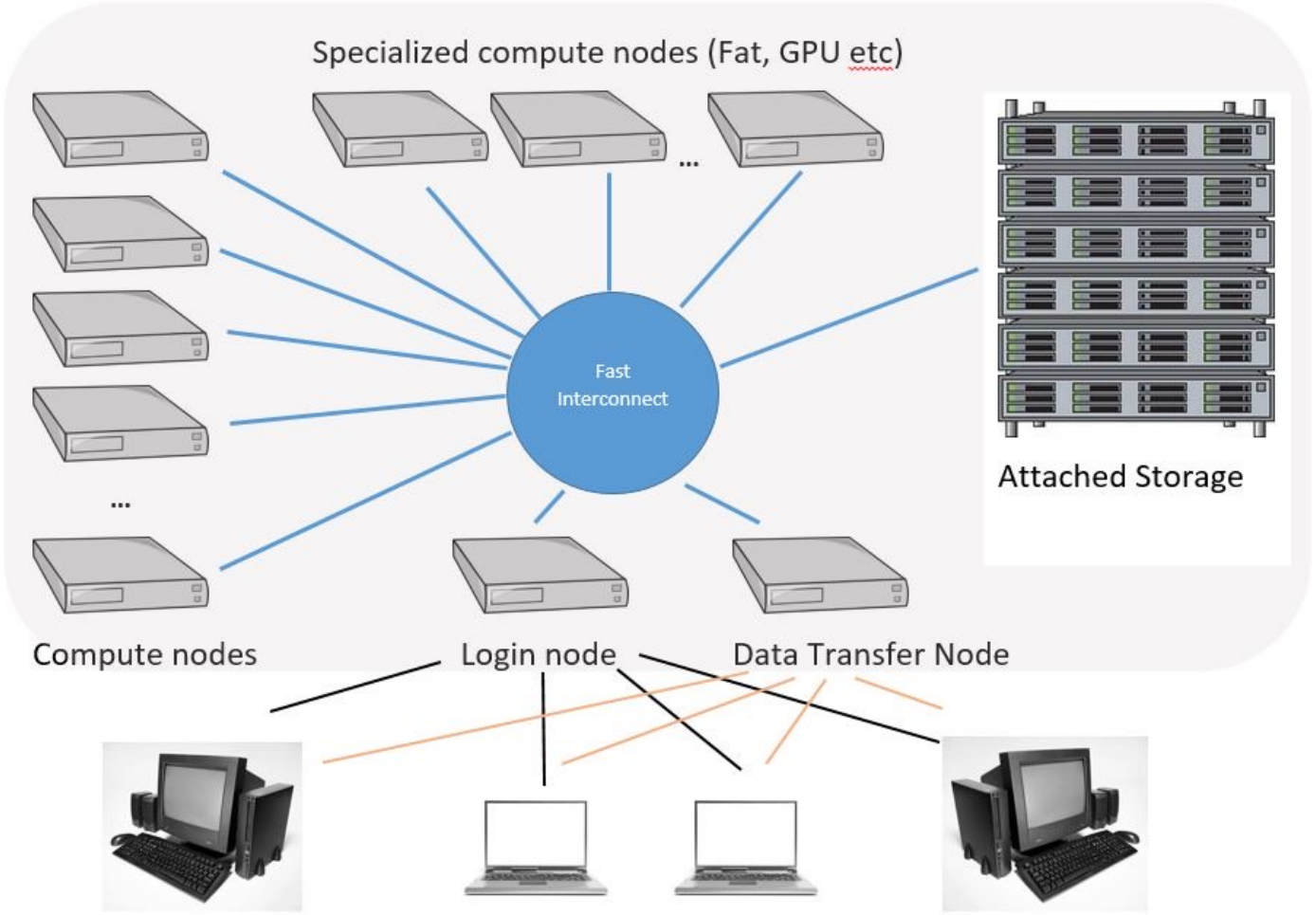


What is a High Performance Computing Cluster?

An HPC cluster is a collection of many separate servers (computers), called nodes, which are connected via a **fast** interconnect.

There may be different types of nodes for different types of tasks.

All cluster nodes have the same components as a laptop or desktop: CPU cores, memory and disk space. The difference between personal computer and a cluster node is in quantity, quality and power of the components.



[Source: https://www.hpc.iastate.edu/guides/introduction-to-hpc-clusters/what-is-an-hpc-cluster]

Node types at UMBC High Performance Computing Facility

- The **CPU cluster** consists of 18 compute nodes with two 24-core Intel Cascade Lake CPUs and 196 GB of memory each, 42 compute nodes with two 18-core Intel Skylake CPUs and 384 GB of memory each, and 49 compute nodes with two 8-core Intel Ivy Bridge CPUs and 64 GB of memory each. This cluster also includes 4 nodes in a develop partition and one interactive node
- The **taki GPU cluster** contains one node with four NVIDIA Tesla V100 GPUs connected by NVLink and 18 hybrid CPU/GPU nodes with two NVIDIA K20 GPUs.
- The **Big Data cluster** consists of 8 nodes in taki that are equipped with two 18-core Intel Skylake CPUs, 384 GB of memory, and 48 TB disk space distributed across 12 hard drives each.
- The **ada GPU cluster** is comprised of 13 nodes which each have two 24-core Intel Cascade Lake CPUs and 384 GB of memory. Four of these nodes have eight 2080 Ti GPUs; seven of these nodes have eight RTX 6000 GPUs; and two of these nodes have eight RTX 8000 GPUs with an extra 384 GB of memory each.

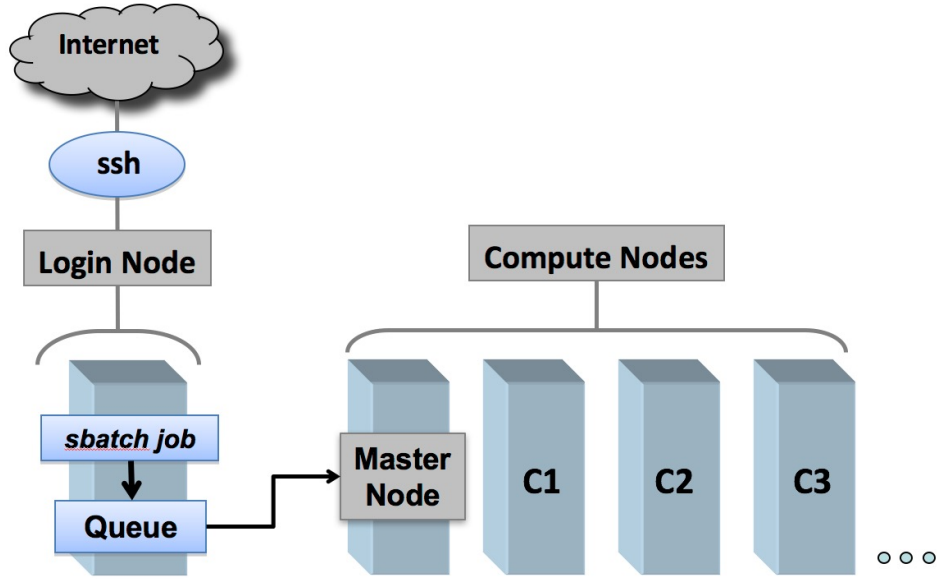
[Source: https://hpcf.umbc.edu/]

Connecting to HPCCs and running code with SLURM

HPCCs generally require users to request and account to access the system, and specific login information can be found in the online HPCC documentation (https://hpcf.umbc.edu/usage-rules/)

The Slurm Workload Manager, formerly known as Simple Linux Utility for Resource Management, or simply Slurm, is a free and open-source job scheduler for Linux and Unix-like kernels, used by many of the world's supercomputers and computer clusters.

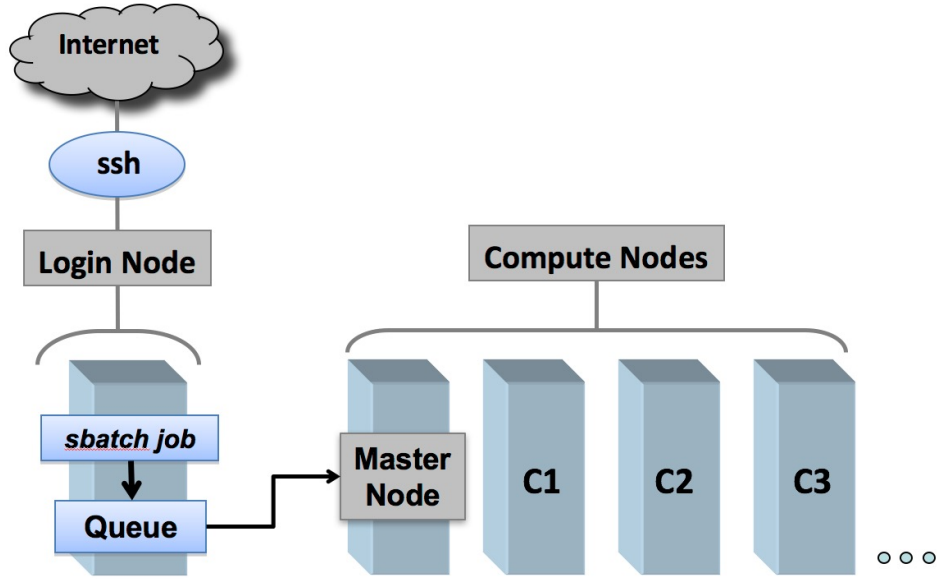
Wikipedia



Connecting to HPCCs and running code with SLURM

HPCCs generally require users to request and account to access the system, and specific login information can be found in the online HPCC documentation (https://hpcf.umbc.edu/usage-rules/)

1. Login to the login node with user credentials
2. Upload code to server (in script format)
3. Submit script to SLURM queue ("job")
4. Wait for SLURM to assign your job to a node
5. The assigned node will automatically run your code
6. Download processed data once the job is complete

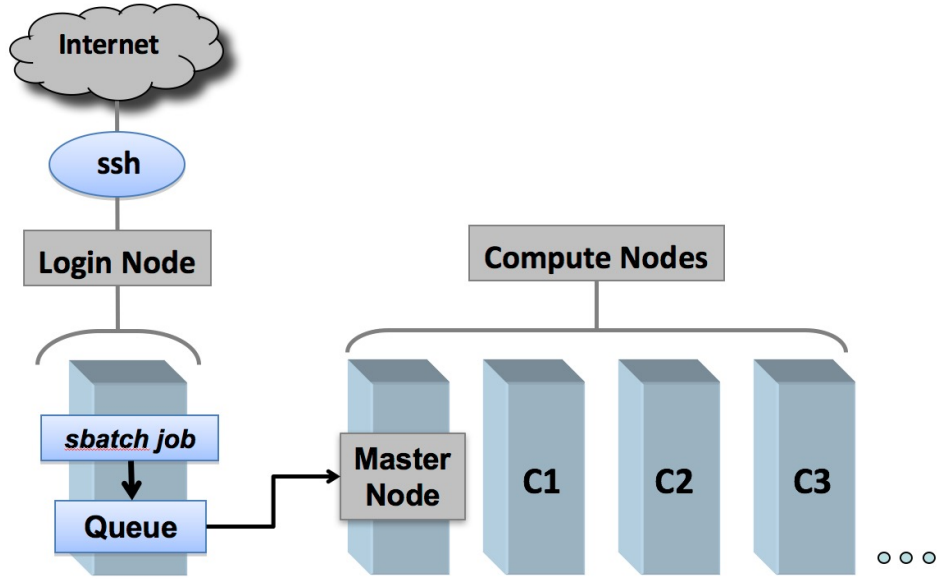


A ***script*** is a program or sequence of instructions that is interpreted or carried out by another program

Connecting to HPCCs and running code with SLURM

HPCCs generally require users to request and account to access the system, and specific login information can be found in the online HPCC documentation (https://hpcf.umbc.edu/usage-rules/)

1. Login to the login node with user credentials
2. Upload code to server (in script format)
3. Submit script to SLURM queue ("job")
4. Wait for SLURM to assign your job to a node
5. The assigned node will automatically run your code
6. Download processed data once the job is complete



Basic scheduling is done by first-come-first-served, with priorities adjusted by factors like job size and time waited. The scheduler used currently is the software SLURM developed at Lawrence Livermore National Laboratory. SLURM prioritizes jobs based on a multi-factor plugin, which weighs several factors together to determine jobs' priority.

Connecting to HPCCs

HPCCs generally require users to request and account to access the system, and specific login information can be found in the online HPCC documentation (https://hpcf.umbc.edu/usage-rules/)

Running a program on taki is different than running one on a standard workstation. When we log into the cluster, we are interacting with the login node. But we would like our programs to run on the compute nodes, which is where the real computing power of the cluster is. We will walk through the processes of running serial and parallel code on the cluster, and then later discuss some of the finer details. This page uses the code examples from the compilation tutorial. Please download and compile those examples first, before following the run examples here.

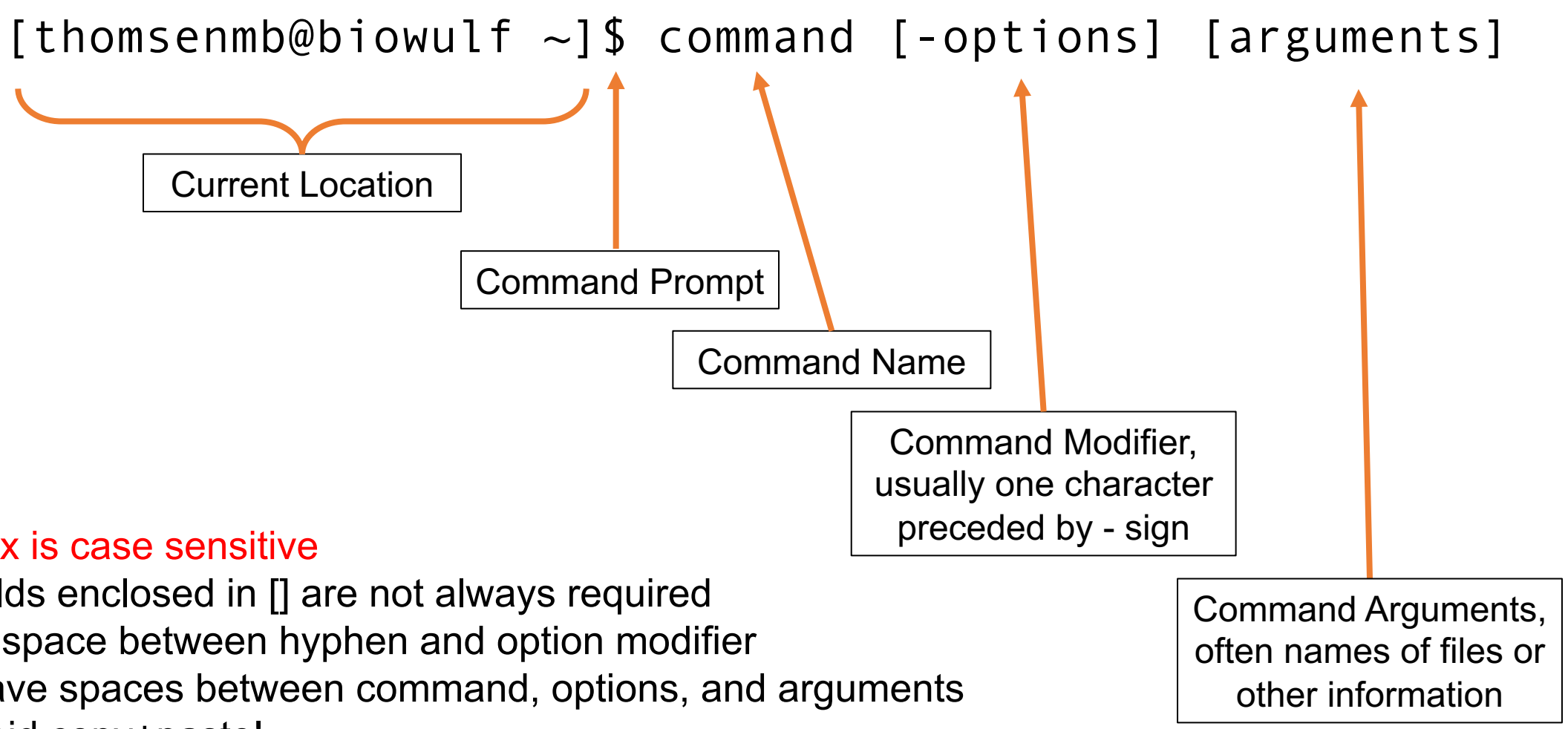
On taki, jobs must be run on the compute nodes of the cluster. You cannot execute jobs directly on the compute nodes yourself; you must request the cluster's batch system do it on your behalf. To use the batch system, you will submit a special script which contains instructions to execute your job on the compute nodes. When submitting your job, you specify a partition (group of nodes, e.g., develop for testing or, for instance, batch for production) and a QOS (a classification that determines what kind of resources your job will need). Your job will wait in the queue until it is "next in line", and free processors on the compute nodes become available. Which job is "next in line" is determined by the scheduling rules of the cluster. Once a job is started, it continues to run until it either completes (with or without error) or reaches its time limit, in which case it is terminated by the scheduler.

During the runtime of your job, your instructions will be executed across the compute nodes. These instructions will have access to the resources of the nodes on which they are running. Notably, the memory, processors, and local disk space (/scratch space). Note that the /scratch space is cleared after every job terminates.

The batch system (also called the scheduler or work load manager) used on taki is called SLURM, which is short for Simple Linux Utility for Resource Management.

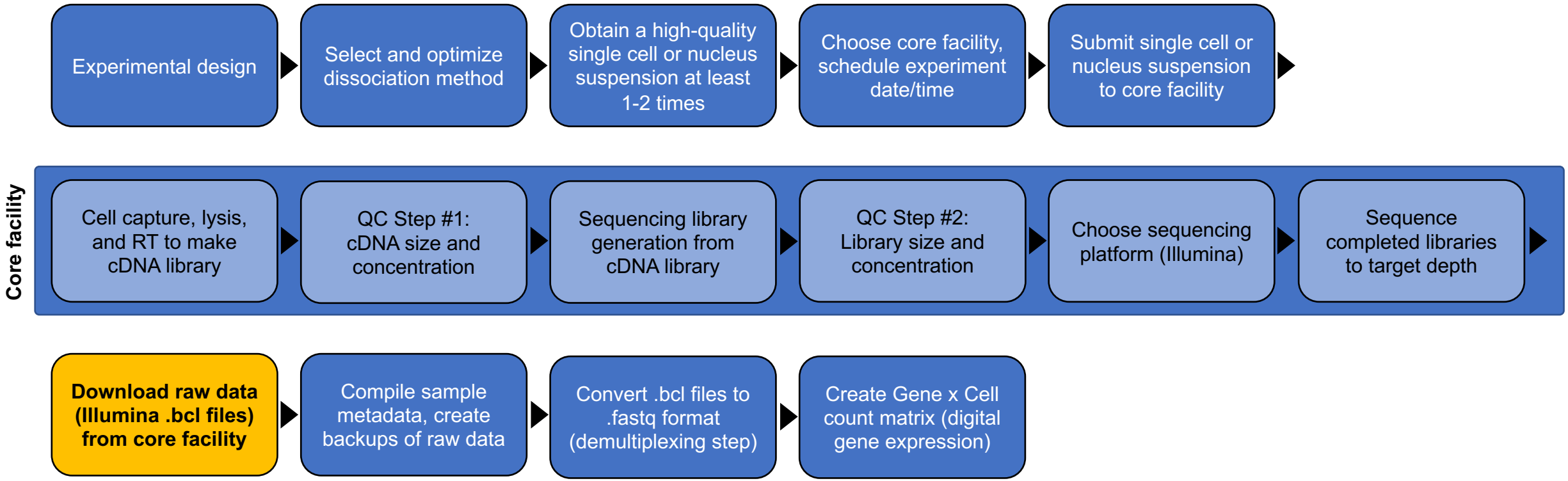
[Source: https://hpcf.umbc.edu/cpu/how-to-run-programs-on-taki/]

Introduction to the Linux/Unix Command Line



- Unix is case sensitive
- Fields enclosed in [] are not always required
- No space between hyphen and option modifier
- Leave spaces between command, options, and arguments
- Avoid copy+paste!

Single Cell Experiment Outline



Goal: transform raw sequencing data into a “count matrix”

Cell:	1	2	···	*N*
GENE 1	1	2		14
GENE 2	4	27		8
GENE 3	0	0		1
⋮	⋮	⋮		⋮
GENE M	6	2		0

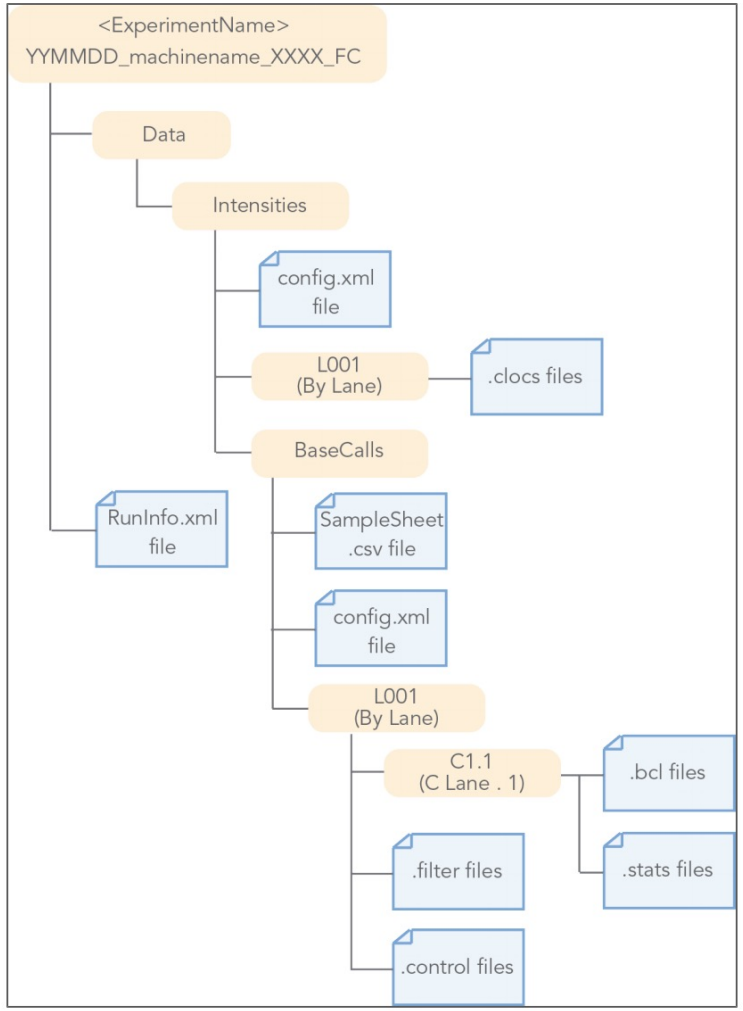
Data Formats, Handling, & Storage – Illumina .bcl files

.bcl files = “base call” file

Typical data structure shown on right →

(note: this structure may vary slightly depending on the sequencer/version/etc)

Figure 3 Bcl Conversion Input Files



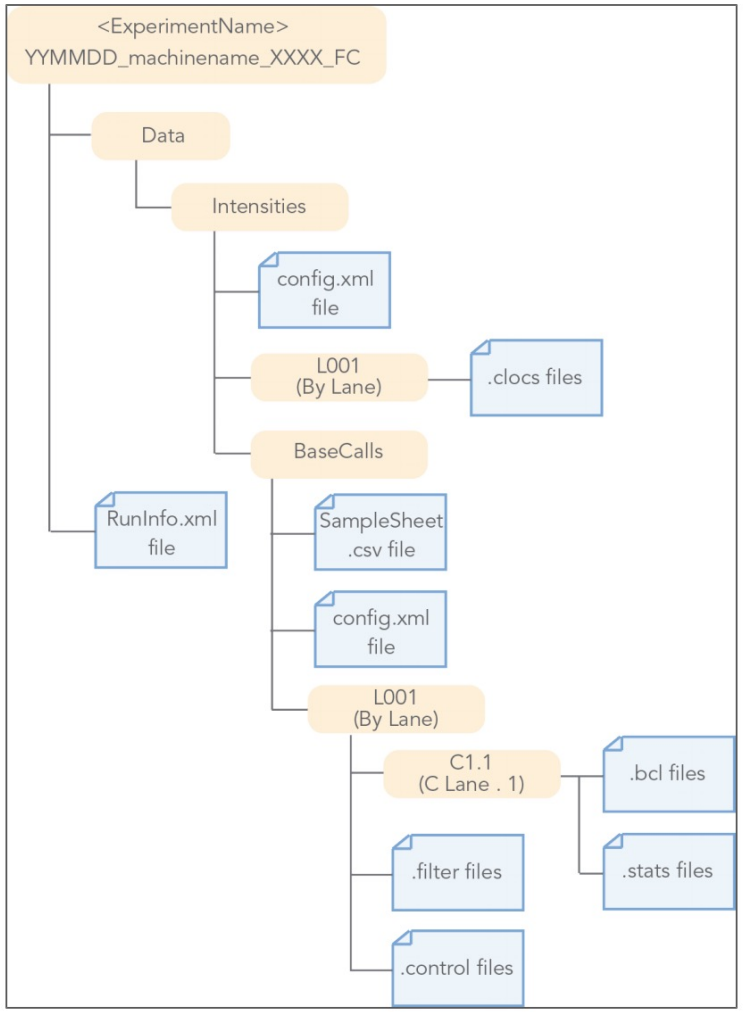
[Source: https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/bcl2fastq/bcl2fastq_letterbooklet_15038058brpmi.pdf]

Data Formats, Handling, & Storage – Illumina .bcl files

BCL File output from NextSeq500 Run

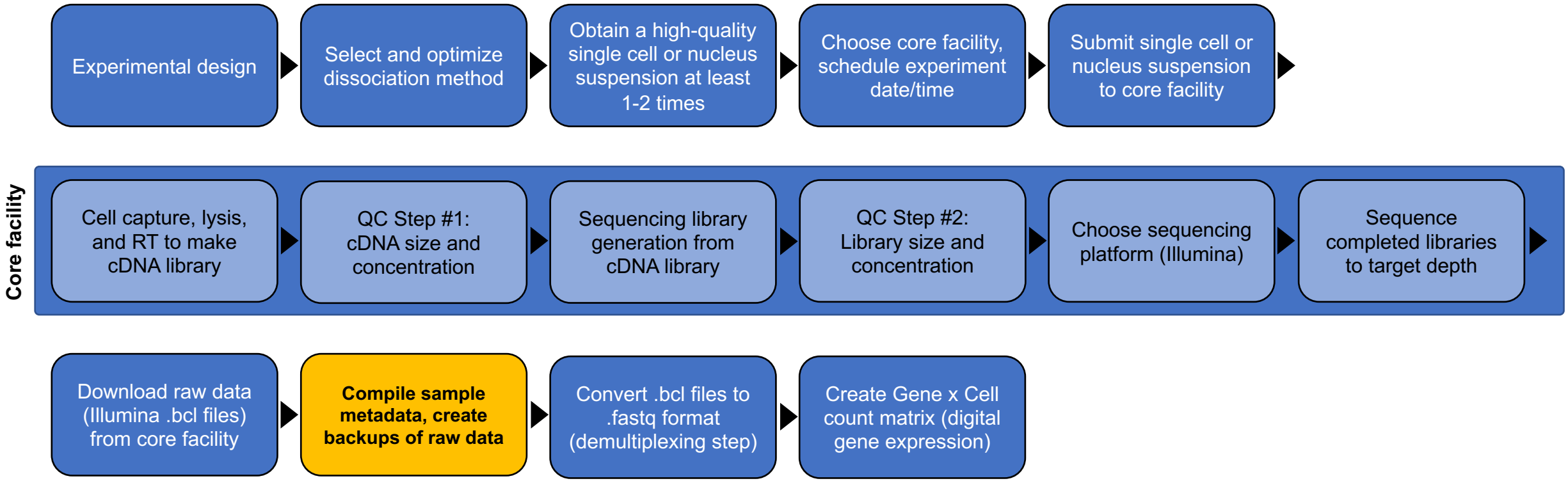
```
[thomsenmb@helix 20210324_nextseq_run]$ ls -l
total 263
drwxr-x--- 3 thomsenmb thomsenmb  4096 May 19 12:55 Data
drwxr-x--- 3 thomsenmb thomsenmb  4096 May 19 12:57 InterOp
drwxr-x--- 2 thomsenmb thomsenmb  4096 May 19 12:57 Logs
-rw-r----- 1 thomsenmb thomsenmb    46 May 19 12:57 RTAComplete.txt
drwxr-x--- 2 thomsenmb thomsenmb  4096 May 19 12:57 RTALogs
-rw-r----- 1 thomsenmb thomsenmb    36 May 19 12:55 RTARead1Complete.txt
-rw-r----- 1 thomsenmb thomsenmb    37 May 19 12:55 RTARead2Complete.txt
-rw-r----- 1 thomsenmb thomsenmb    36 May 19 12:57 RTARead3Complete.txt
-rw-r----- 1 thomsenmb thomsenmb  2567 May 19 12:57 Ruchi_03221_203983800.json
-rw-r----- 1 thomsenmb thomsenmb   924 May 19 12:57 RunCompletionStatus.xml
-rw-r----- 1 thomsenmb thomsenmb 28569 May 19 12:55 RunInfo.xml
-rw-r----- 1 thomsenmb thomsenmb 26364 May 19 12:55 RunParameters.xml
[thomsenmb@helix 20210324_nextseq_run]$
```

Figure 3 Bcl Conversion Input Files



[Source: https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/bcl2fastq/bcl2fastq_letterbooklet_15038058brpmi.pdf]

Single Cell Experiment Outline



Sample metadata records

Best practice: keep records accurate, detailed, And up-to-date

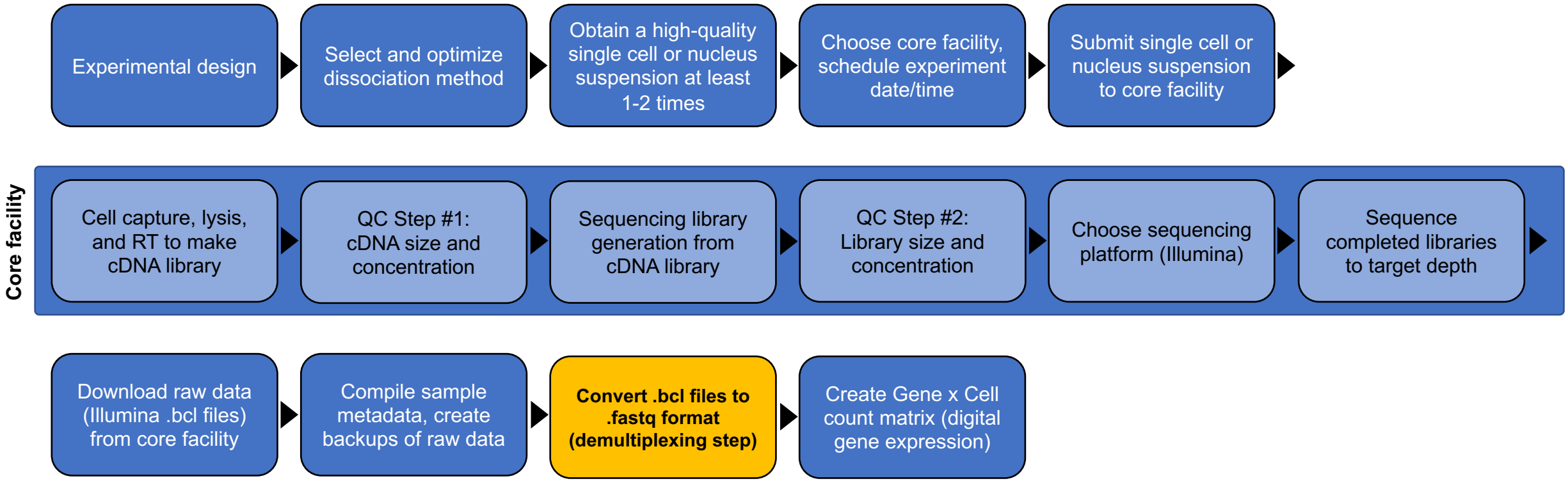
	A	B	C	D	E	F	G	H	I	J
1	**10X Index**	**Sample ID**	**Sample Name**	**Project**	**Seq Run Date**	**Chemistry**	**Sample Tissue**	**Sample Treatment**	**Sample Genotype**	**Sample Prep**
2	A1	A1	SCN	SCN		V2	SCN	-	WT	
3	A2	A2	SCN	SCN		V2	SCN	-	WT	
4	A3	JZ-A3	CTRL_PFC_20190117	PFC	20190130	V2	PFC	-		
5	A4	JZ-A4	CTRL_PFC_20190117	PFC	20190130	V2	PFC	-		
6	A5	JZ-A5	DTA_PFC_20190124	PFC	20190130	V2	PFC	-		
7	A6	JZ-A6	DTA_PFC_201090311	PFC	20190405	V2	PFC	-		
8	A7	JZ-A7	DTA_MC_20190311	PFC	20190423	V2	MC	-		
9	A8	JZ-A8	DTA_PFC_20190319	PFC	20190405	V2	PFC	-		
10	A9	JZ-A9	CTRL_MC	PFC	not run	V2	MC	-		
11	A10	JZ-A10	CTRL_PFC_20190418	PFC	20190423	V2	PFC	-		
12	A11	JZ-A11	CTRL_MC_20190418	PFC	20190423	V2	MC	-		
13	A12	MT08	SCN_LP_CT14	SCN	20190726	V3	SCN	LP CT14	WT	
14	B1	-	PHB_20181003	PHB	20190130	V2	PHB	-	C57Bl/6J	
15	B2	-	PHB_20181003	PHB	20190130	V2	PHB	-	C57Bl/6J	
16	B3	-	PHB_20190225	PHB	20190405	V2	PHB	-	C57Bl/6J	
17	B4	MT01	SCN_LP_CT6	SCN	20190510	V3	SCN	LP CT6	WT	
18	B5	MT02	SCN_LP_CT6	SCN	20190510	V3	SCN	LP CT6	WT	
19	B6	MT05	SCN_LP_CT14	SCN	20190716	V3	SCN	LP CT14	WT	
20	B7	MT06	SCN_LP_CT14	SCN	20190716	V3	SCN	LP CT14	WT	
21	B8	MT07	DTA_CTRL_CT14	DTA-SCN	20190716	V3	SCN	NLP CT14	Opn4+/+	
22	B9	MT09	DTA_CT14	DTA-SCN	20191111	V3	SCN	NLP CT14	Opn4mDTA/mDTA	Frozen Nuclei
23	B10	MT10	SCN_LP_CT22	SCN	20190726	V3	SCN	LP CT 22	WT	

Data Formats, Handling, & Storage – tarballs

In computing, **tar** is a computer software utility for collecting many files into one archive file, often referred to as a **tarball**, for distribution or backup purposes. The name is derived from "**t**ape **ar**chive", as it was originally developed to write data to sequential I/O devices with no file system of their own. The archive data sets created by tar contain various file system parameters, such as name, timestamps, ownership, file-access permissions, and directory organization.

[Source: https://en.wikipedia.org/wiki/Tar_(computing)]

Single Cell Experiment Outline



Data Formats, Handling, & Storage – .fastq files

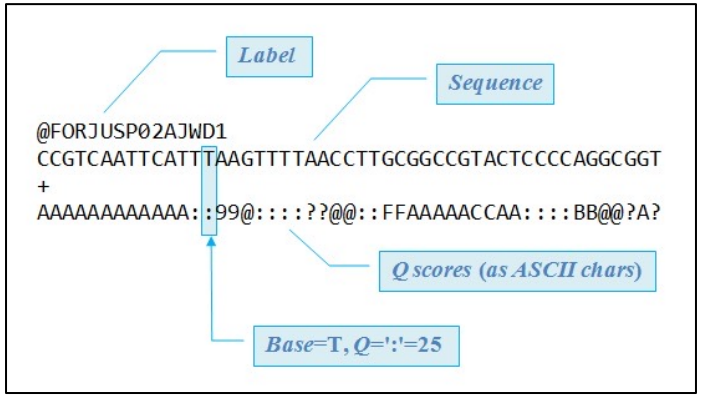
FASTQ format

From Wikipedia, the free encyclopedia

FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. Both the sequence letter and quality score are each encoded with a single ASCII character for brevity.

It was originally developed at the Wellcome Trust Sanger Institute to bundle a FASTA formatted sequence and its quality data, but has recently become the *de facto* standard for storing the output of high-throughput sequencing instruments such as the Illumina Genome Analyzer.[1]

"fasta + quality"



[Source: https://drive5.com/usearch/manual/fastq_files.html]

10X Genomics Cell Ranger Software

What is Cell Ranger?

Cell Ranger is a set of analysis pipelines that process Chromium single-cell data to align reads, generate feature-barcode matrices, perform clustering and other secondary analysis, and more. Cell Ranger includes four pipelines relevant to the 3' Single Cell Gene Expression Solution and related products:

- **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 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 on the same GEM well. `cellranger count` also processes Feature Barcode data alongside Gene Expression reads.
- **cellranger aggr** aggregates outputs from multiple runs of `cellranger count`, normalizing those runs to the same sequencing depth and then recomputing the feature-barcode matrices and analysis on the combined data. The `aggr` pipeline can be used to combine data from multiple samples into an experiment-wide feature-barcode matrix and analysis.

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger

Generating .fastq files from .bcl files using Cellranger mkfastq

Required input

.bcl files
samplesheet
output directory name

Output

Demultiplexed .fastq files

mkfastq.sh

```
cellranger mkfastq --run=20210324_nextseq_run \
        --samplesheet=03242021_samplesheet.csv \
        --localcores=$SLURM_CPUS_PER_TASK \
        --output-dir demultiplexed \
        --localmem=50
```

- Path to .bcl directory
- Path to samplesheet
- Number of CPUs for cellranger to use
- Name of the output directory (user choice)
- Amount of local memory (RAM) for cellranger to use

Generating .fastq files from .bcl files using Cellranger – Sample Sheet Info

<run_date>_samplesheet.csv

	A	B	C
1	lane	sample	index
2	*	MT39	SI-GA-B12
3	*	MT40	SI-GA-G3
4	*	MT41	SI-GA-D9
5			

Lane: specify which lanes of the Illumina sequencer contained your samples. This sample was sequenced on the NextSeq500. Though there are 4 physical lanes on a NextSeq500 flow cell, samples are run on all lanes at once – to specify all lanes we use the ”wildcard” symbol: “*”

Sample: the name of your sample. This is decided by the user, and will be the name of the directory containing fastq files for that sample

Index: the 10X Genomics library index identifier for the sample. (see link below for exact names – note that the core usually just gives the final alphanumeric code, but “SI-GA-” must be added or the program will fail.

NOTE: the sample sheet must be saved in .csv format

10X index IDs and sequences: https://support.10xgenomics.com/single-cell-gene-expression/index/doc/specifications-sample-index-sets-for-single-cell-3

Cellranger mkfastq Output

```
[[thomsenmb@biowulf sc-workshop-day1]$ ls -l
total 32143365
-rw-r--r--. 1 thomsenmb thomsenmb         450 Jun  7 10:21 03242021_mkfastq.sh
-rw-r--r--. 1 thomsenmb thomsenmb          69 Jun  7 10:19 03242021_samplesheet.csv
drwxr-x---. 6 thomsenmb thomsenmb        4096 May 19 12:57 20210324_nextseq_run
-rw-r-----. 1 thomsenmb thomsenmb 32914588751 Jun  7 09:51 20210324_nextseq_run.tar.gz
drwxr-xr-x. 5 thomsenmb thomsenmb        4096 Jun  7 11:27 demultiplexed
drwxr-xr-x. 4 thomsenmb thomsenmb        4096 Jun  7 11:28 H3MGYBGXG
-rw-r--r--. 1 thomsenmb thomsenmb         542 Jun  7 10:20 MT39_count.sh
-rw-r--r--. 1 thomsenmb thomsenmb        5875 Jun  7 11:28 slurm-16559152.out
[thomsenmb@biowulf sc-workshop-day1]$
```

.fastq files are in this directory

The output SLURM log is here (# = job ID)

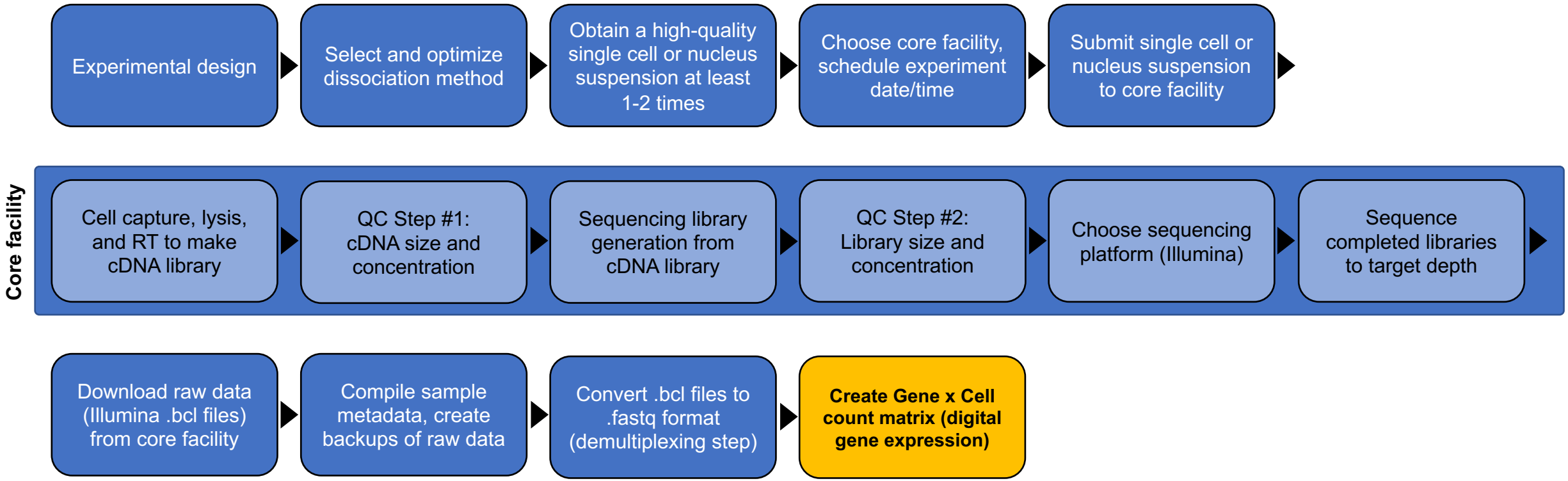
Cellranger mkfastq Output

```
[[thomsenmb@biowulf sc-workshop-day1]$ cd demultiplexed/H3MGYBGXG/
[[thomsenmb@biowulf H3MGYBGXG]$ pwd
 /data/thomsenmb/sc-workshop-day1/demultiplexed/H3MGYBGXG
[[thomsenmb@biowulf H3MGYBGXG]$ ls -l
 total 4
 drwxr-xr-x. 2 thomsenmb thomsenmb 4096 Jun  7 11:28 MT39
 -rw-r--r--. 1 thomsenmb thomsenmb  542 Jun  7 10:20 MT39_count.sh
 drwxr-xr-x. 2 thomsenmb thomsenmb 4096 Jun  7 11:28 MT40
 drwxr-xr-x. 2 thomsenmb thomsenmb 4096 Jun  7 11:28 MT41
[[thomsenmb@biowulf H3MGYBGXG]$ cd MT39
[[thomsenmb@biowulf MT39]$ ls -l
 total 15344000
 -rw-r--r--. 1 thomsenmb thomsenmb  318180748 Jun  7 11:28 MT39_S1_L001_I1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 1029046035 Jun  7 11:28 MT39_S1_L001_R1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 2579144689 Jun  7 11:28 MT39_S1_L001_R2_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb  315300988 Jun  7 11:28 MT39_S1_L002_I1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 1011234377 Jun  7 11:28 MT39_S1_L002_R1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 2547454575 Jun  7 11:28 MT39_S1_L002_R2_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb  324079329 Jun  7 11:28 MT39_S1_L003_I1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 1043748817 Jun  7 11:28 MT39_S1_L003_R1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 2620726546 Jun  7 11:28 MT39_S1_L003_R2_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb  319709931 Jun  7 11:28 MT39_S1_L004_I1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 1026157574 Jun  7 11:28 MT39_S1_L004_R1_001.fastq.gz
 -rw-r--r--. 1 thomsenmb thomsenmb 2576715011 Jun  7 11:28 MT39_S1_L004_R2_001.fastq.gz
```

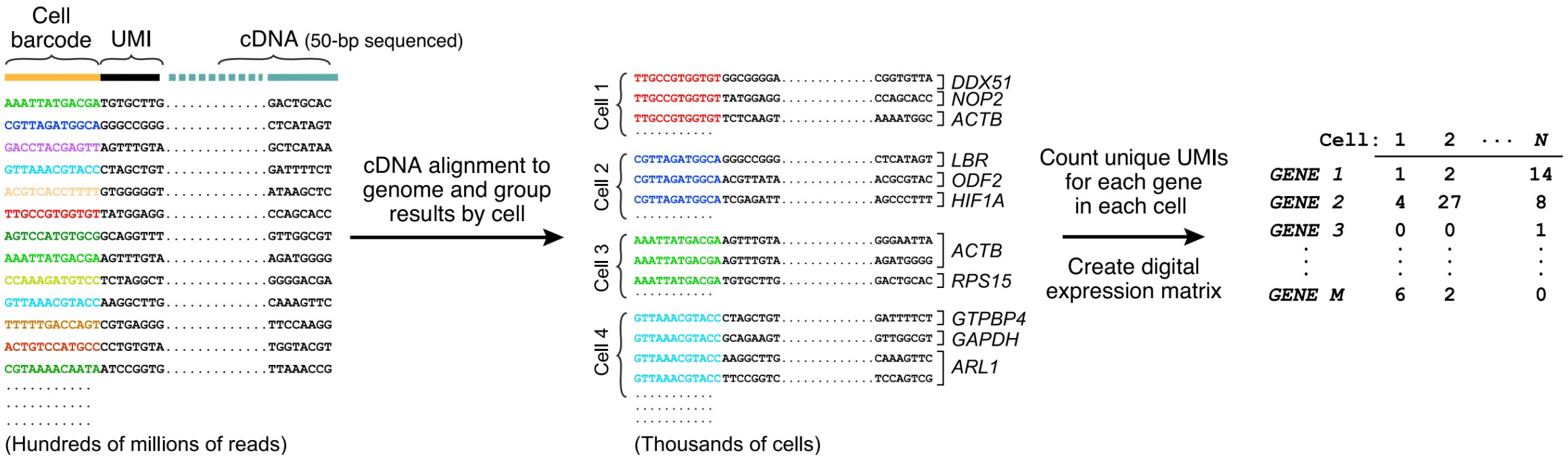
These directories contain sample-specific .fastq files

.fastq files for sample ‘MT39’

Single Cell Experiment Outline

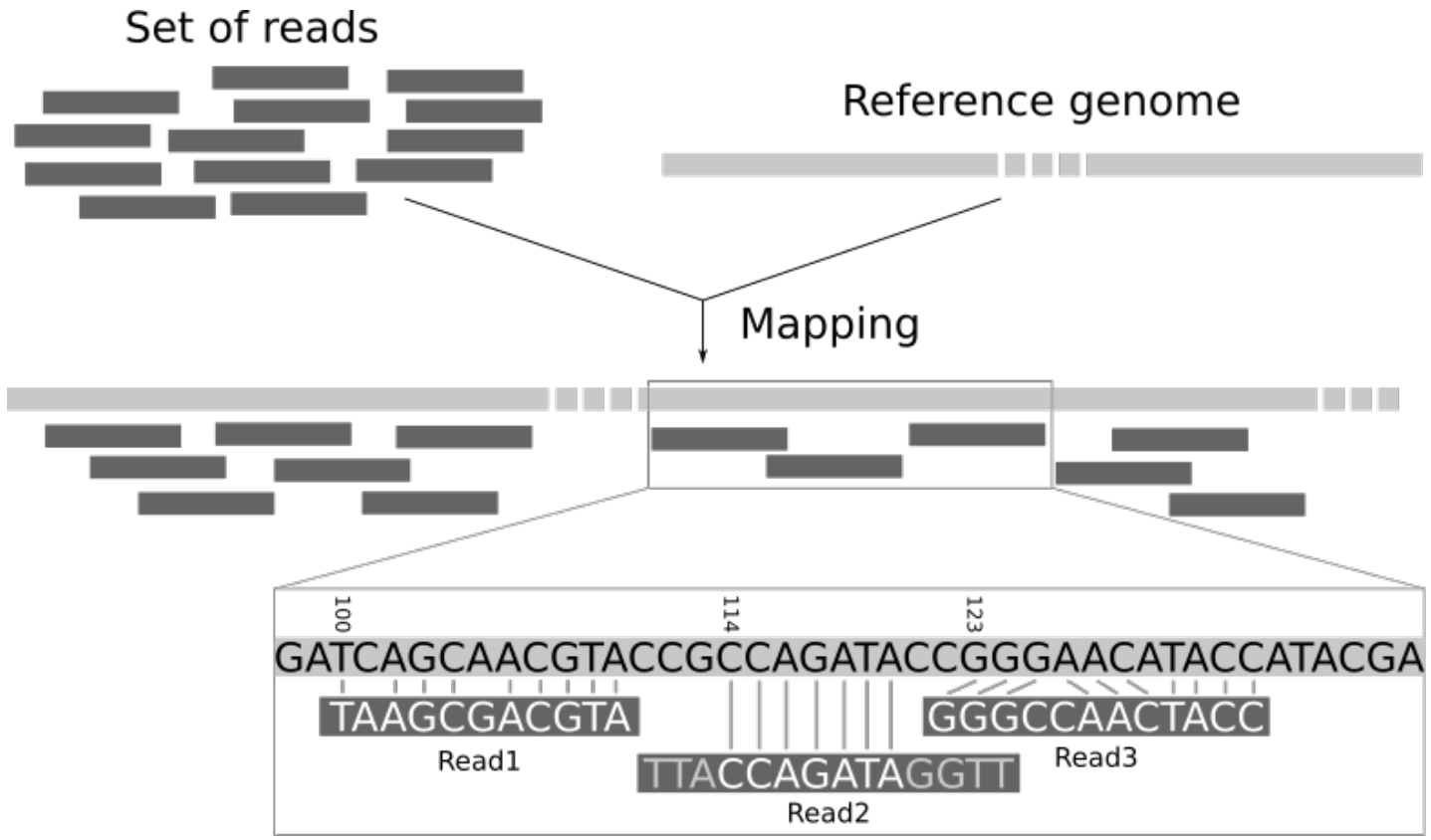


Generating DGE matrices from .fastq files using Cellranger count



[Source: Macosko et al., 2015 *Cell*]

Generating DGE matrices from .fastq files using Cellranger count



[Source: https://galaxyproject.github.io/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html]

Generating DGE matrices from .fastq files using Cellranger count

Required input

.fastq files
reference genome
output directory

→

Output

Digital gene expression matrix
HTML experiment summary

count.sh

```
mkdir MT39_out

cellranger count --id MT39_out \
        --fastqs MT39 \
        --transcriptome=$CELLRANGER_REF/refdata-gex-mm10-2020-A \
        --localcores=$SLURM_CPUS_PER_TASK \
        --localmem=200 \
        [redacted]
        --include-introns
```

Notes

Cellranger requires a large amount of RAM, recommend using at least 50GB

Cellranger takes several hours to complete (typically 3-6 hours)

Cellranger requires a *preexisting* output directory

Biowulf has 10X reference genomes available at the $CELLRANGER_REF environment variable

For nucleus libraries, use the –include-introns option to map your library to pre-mRNA

Generating DGE matrices from .fastq files using Cellranger count

Required input

.fastq files
reference genome
output directory

Output

Digital gene expression matrix
HTML experiment summary

count.sh

```
mkdir MT39_out

cellranger count --id MT39_out \
        --fastqs MT39 \
        --transcriptome=$CELLRANGER_REF/refdata-gex-mm10-2020-A \
        --localcores=$SLURM_CPUS_PER_TASK \
        --localmem=200 \
        [redacted]
        --include-introns
```

Name of output directory

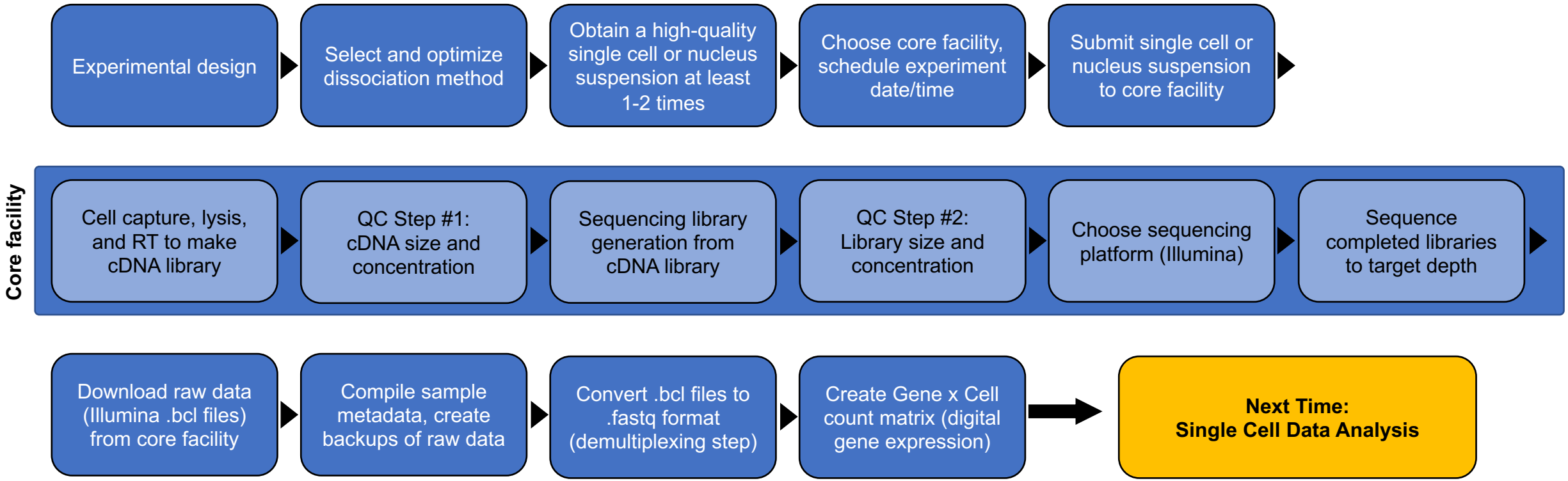
Name of .fastq file directory

Path to reference genome

Desired CPU and RAM

Optional – include for nucleus preparations

Single Cell Experiment Outline



End of Day 1