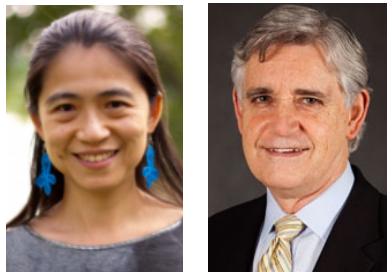


NextGen sequencing and high-performance computing



QB Bootcamp, Day 2
Wednesday, 6 September 2023
1:00pm - 1:30pm

The Stillman uses high-throughput DNA sequencing to study the dynamics of DNA replication initiation and progression



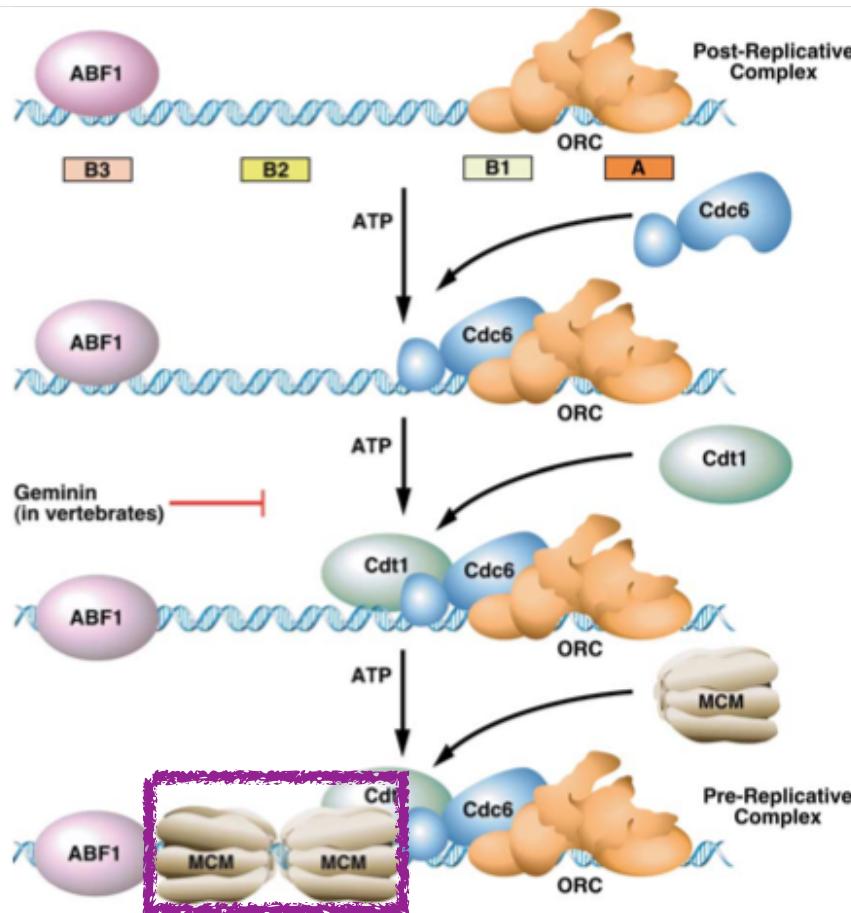
Domain within the helicase subunit Mcm4 integrates multiple kinase signals to control DNA replication initiation and fork progression

Yi-Jun Sheu^a, Justin B. Kinney^a, Armelle Lengronne^b, Philippe Pasero^b, and Bruce Stillman^{a,1}

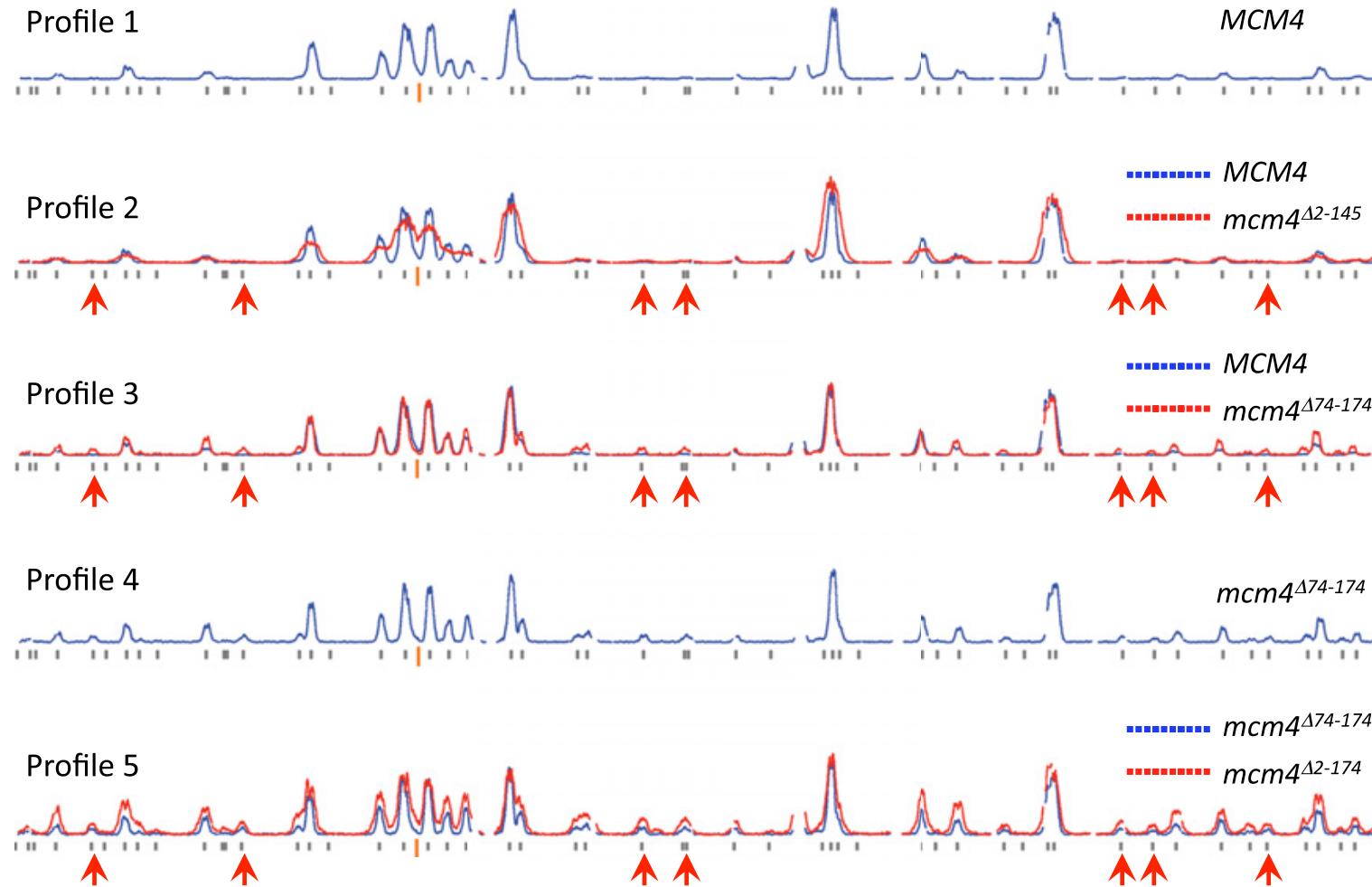
PNAS | Published online April 16, 2014 | E1899–E1908



S. cerevisiae



Here are some examples of the published replication profiles

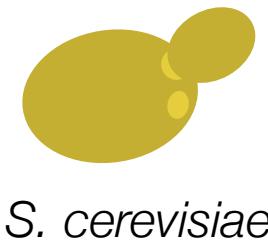


~300 separate loci direct DNA replication initiation in *Saccharomyces cerevisiae*

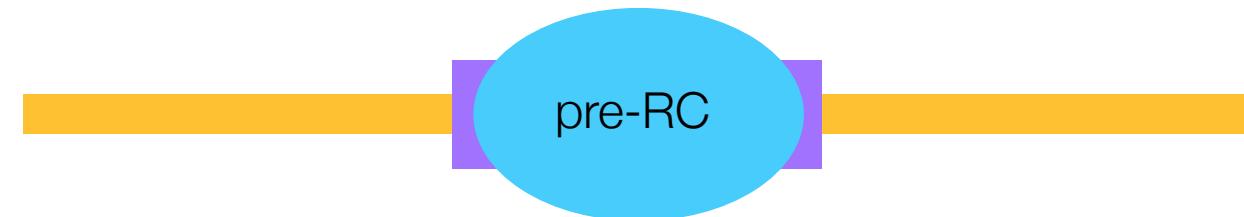
ARS: autonomously replicating sequence

— old ssDNA

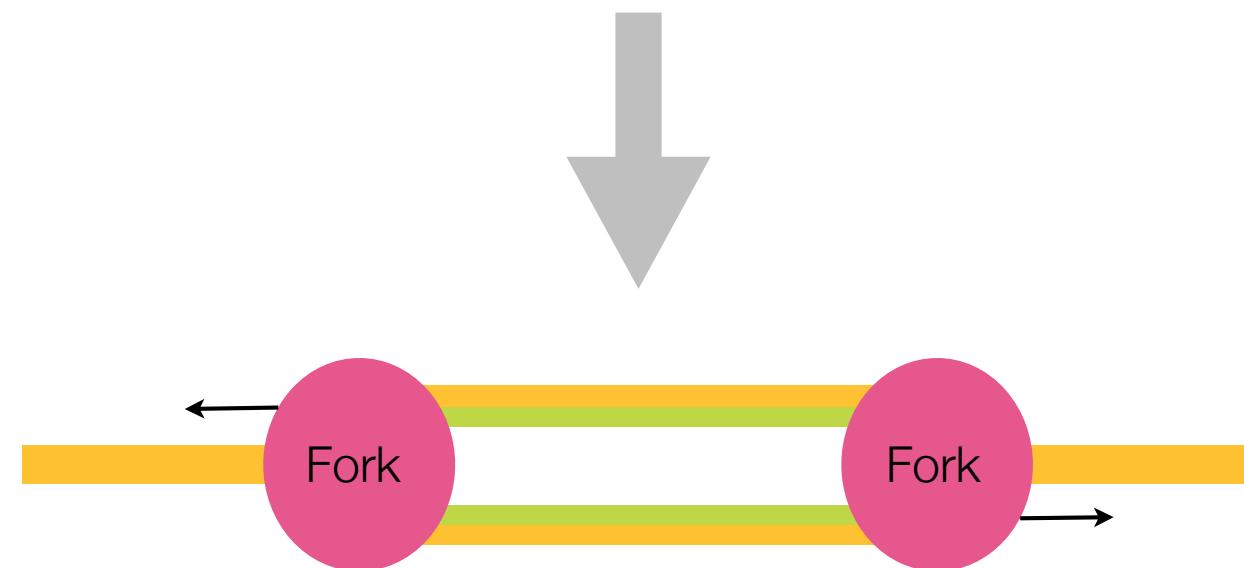
— new ssDNA



G1 phase



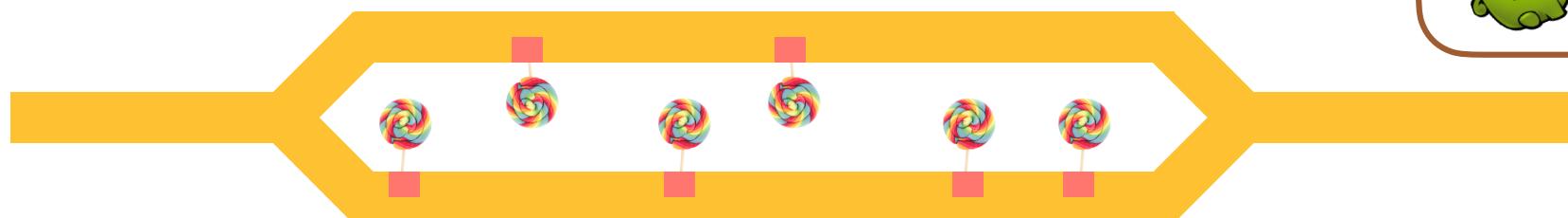
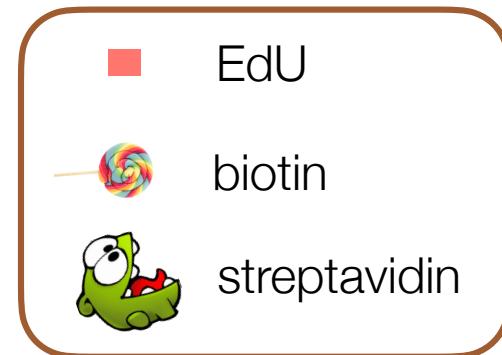
S phase



Newly replicated DNA can be isolated using an EdU pull-down assay

DNA of cells arrested in G1 with a-factor

Release cells into S-phase
EdU incorporation during replication

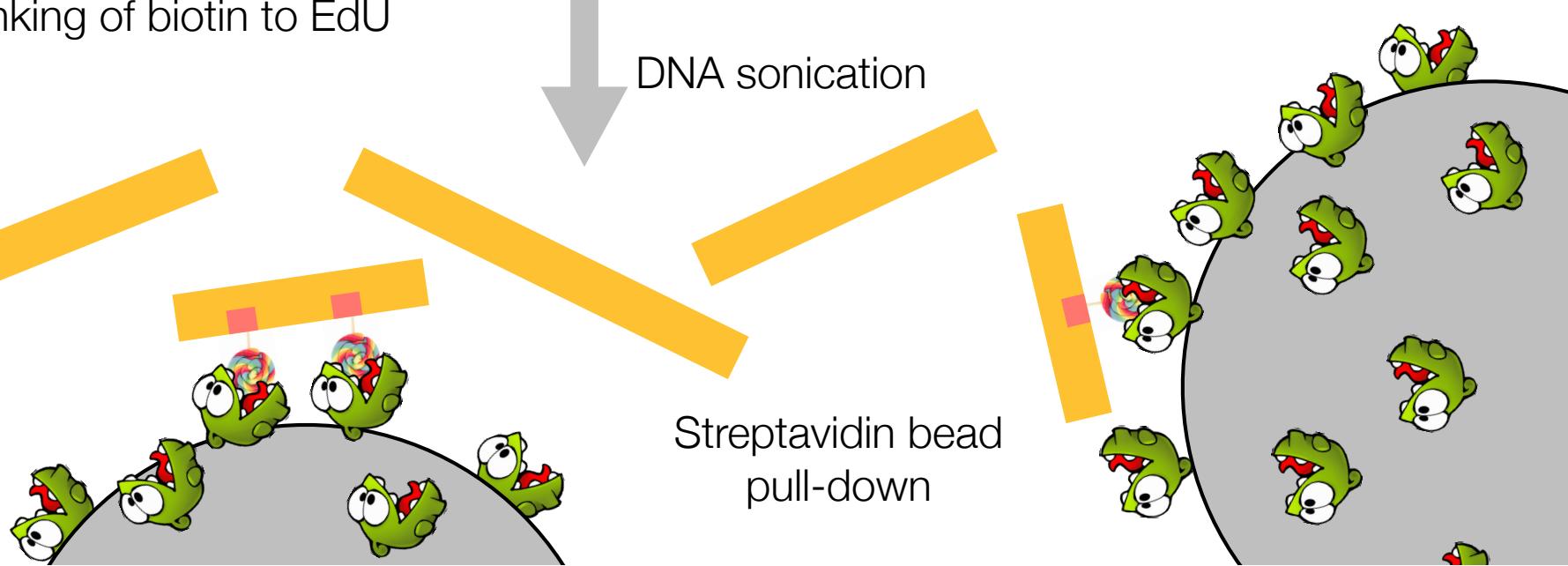


Click-iT linking of biotin to EdU

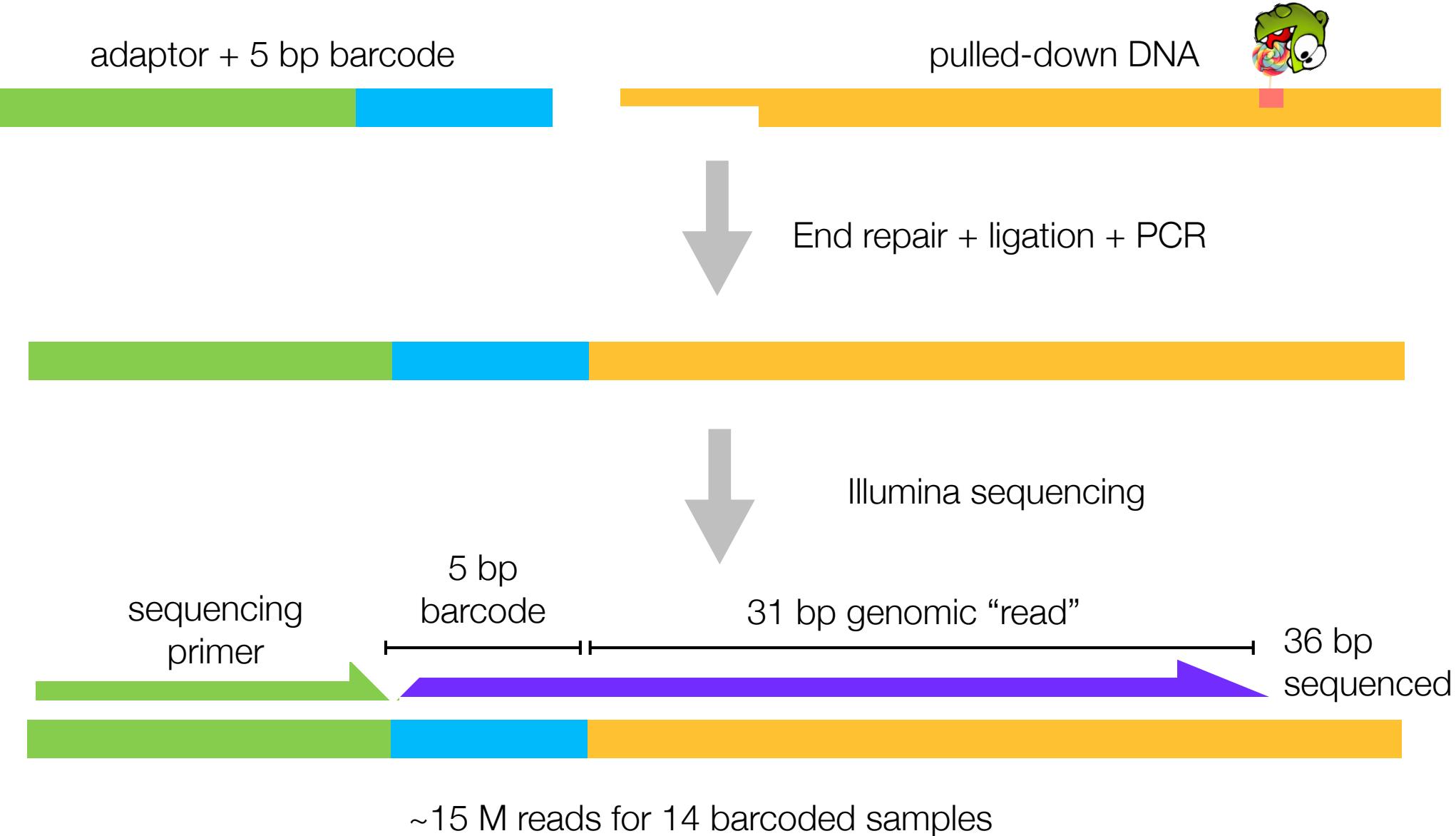
DNA sonication



Streptavidin bead pull-down



Sequencing of pulled-down DNA allows replication to be mapped genome-wide



We will analyze FASTQ read files from 4 different samples

There are four FASTQ files in the reads/ directory

Each FASTQ files is ~60-100 megabytes

```
[[jkinstructor@bamdev1 ~]$ ls
elzar_exercise  elzar_exercise.tar.gz
[[jkinstructor@bamdev1 ~]$ ls -lah elzar_exercise/reads/
total 302M
drwxr-s---  2 jkinstructor wsbs students  4.0K Sep  2 07:05 .
drwxr-sr-x  6 jkinstructor wsbs students  4.0K Sep  2 06:34 ..
-rw-r----- 1 jkinstructor wsbs students 70M Aug 30 2016 A1.fastq
-rw-r----- 1 jkinstructor wsbs students 97M Aug 30 2016 B1.fastq
-rw-r----- 1 jkinstructor wsbs students 68M Aug 30 2016 C1.fastq
-rw-r----- 1 jkinstructor wsbs students 69M Aug 30 2016 D1.fastq
[jkinstructor@bamdev1 ~]$
```

This is what a FASTQ file looks like (circa 2009)

read 1

```
[jkinstructor@bamdev1 ~]$ cd ~/elzar_exercise/reads/
[jkinstructor@bamdev1 reads]$ head -n 20 A1.fastq
@HANNIBAL_0056:7:1:9620:1049#0/1
GTGGTTAGTATATGGTGCAAAAGTGGTATAA
+HANNIBAL_0056:7:1:9620:1049#0/1
ggggggaeadffffccdfaaaaefgfgggg
@HANNIBAL_0056:7:1:1070:1061#0/1
CGAACACAAAGATCTCGTTCTACTTTTTTG
+HANNIBAL_0056:7:1:1070:1061#0/1
f`[facdddfJdcfaa^c fcf dcfffc]
@HANNIBAL_0056:7:1:4279:1052#0/1
TATCCACTACCGCTATACTGGATTCTGACTC
+HANNIBAL_0056:7:1:4279:1052#0/1
hghhhhhhhhhghghghhhhhfhhhfhhhg
@HANNIBAL_0056:7:1:4413:1064#0/1
AAGAAAACGTGCCACCATTGAGTACATCAAC
+HANNIBAL_0056:7:1:4413:1064#0/1
hhhhhhhhcfffffgghhhhgdhffghfb
@HANNIBAL_0056:7:1:5309:1059#0/1
AGTATACTGTGTATATAATAGATATGGAACG
+HANNIBAL_0056:7:1:5309:1059#0/1
bf`ebfcffcfbdbeac^ cfcdffffdf
[jkinstructor@bamdev1 reads]$
```

read 2

read 3

read 4

read 5

The information
for each read is
split over 4 lines

← @name
← sequence
← +name
← quality scores

The yeast genome is in FASTA format

```
[jkinstructor@bamdev1 ~]$ cd elzar_exercise/genome/  
[jkinstructor@bamdev1 genome]$ head genome.fasta  
>1 ref|NC_001133| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=I] [note=R64-1-1]  
CCACACCACACCCACACACCCCACACACACACACACACACACACACA  
CATCCTAACACTACCCTAACACAGGCCATAATCTAACCCCTGGCCAACCTGTCTCTCAACTT  
ACCCCTCATTACCCCTGCCCTCACTCGTTACCCCTGTCCATTCAACCATAACCACTCCGAAC  
CACCATCCATCCCTCTACTTACTACCACTCACCCACCGTTACCCCTCCAATTACCCATATC  
CAACCCACTGCCACTTACCCATTACCCATTACCATCCACCATGACCTACTCACCATAAC  
TGTCTTCTACCCACCATATTGAAACGCTAACAAATGATCGTAATAACACACACACGTGCT  
TACCTTACCACTTTATACCACCACTGCCATACTCACCCCTCACTGTATACTGATT  
TACGTACGCACACGGATGCTACAGTATATACCATCTCAAACCTTACCCCTACTCTCAGATT  
CACTTCACTCCATGCCCATCTCACTGAATCAGTACCAAATGCACTCACATCATTATG  
[jkinstructor@bamdev1 genome]$
```

Each header line starts with ‘>’

The corresponding sequence follows,
usually split over lines 80bp long

genome.fasta contains sequences #1 to #16, representing the 16 chromosomes

```
[jkinstructor@bamdev1 genome]$ cat genome.fasta | grep '>'  
>1 ref|NC_001133| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=I] [note=R64-1-1]  
>2 ref|NC_001134| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=II] [note=R64-1-1]  
>3 ref|NC_001135| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=III] [note=R64-1-1]  
>4 ref|NC_001136| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=IV] [note=R64-1-1]  
>5 ref|NC_001137| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=V] [note=R64-1-1]  
>6 ref|NC_001138| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=VI] [note=R64-1-1]  
>7 ref|NC_001139| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=VII] [note=R64-1-1]  
>8 ref|NC_001140| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=VIII] [note=R64-1-1]  
>9 ref|NC_001141| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=IX] [note=R64-1-1]  
>10 ref|NC_001142| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=X] [note=R64-1-1]  
>11 ref|NC_001143| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XI] [note=R64-1-1]  
>12 ref|NC_001144| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XII] [note=R64-1-1]  
>13 ref|NC_001145| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XIII] [note=R64-1-1]  
>14 ref|NC_001146| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XIV] [note=R64-1-1]  
>15 ref|NC_001147| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XV] [note=R64-1-1]  
>16 ref|NC_001148| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=XVI] [note=R64-1-1]  
[jkinstructor@bamdev1 genome]$
```

We will map reads to the genome on the cluster,
then analyze the resulting .bed files on our local machines

A1.fastq + genome.fasta



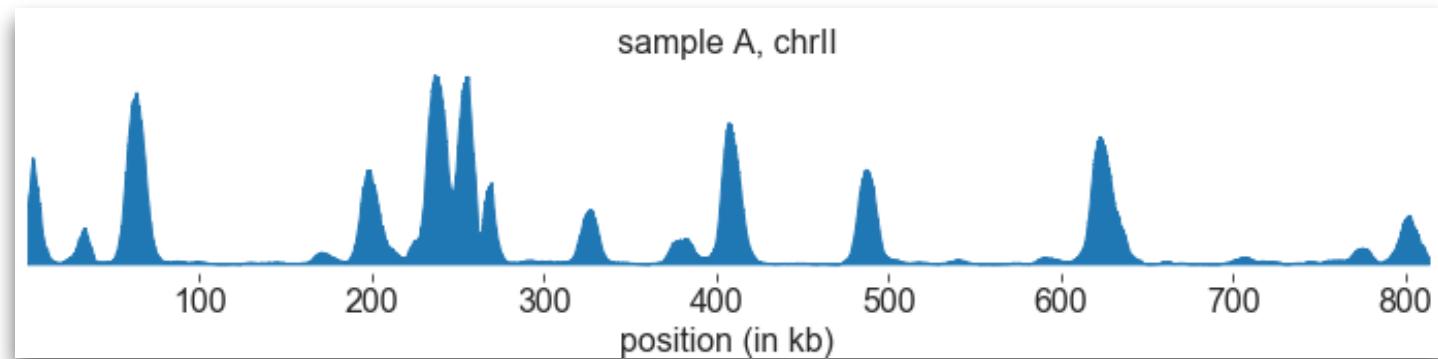
cluster analysis
(bwa + samtools + bedtools)

A1.pileup.bed

```
browser position chrIV:1-1531933
track type=bedGraph visibility=2 name="A1" description="A1"
chrI    1      31     2
chrI    32     62     0
chrI    63     93     1
chrI    94     124    0
chrI    125    155    3
chrI    156    186    0
chrI    187    217    0
chrI    218    248    0
```

chromosome # reads
 window

local analysis
(python)



Elzar is the CSHL's high-performance computer cluster (HPCC)

<http://intranet.cshl.edu/administration/information-technology/hpcc/elzar>

The screenshot shows a web browser window for the CSHL Intranet. The URL in the address bar is <http://intranet.cshl.edu/administration/information-technology/hpcc/elzar>. The page title is "The Essence of Elzar". On the left, there is a sidebar menu under "Information Technology" with links like Home, Divisions (4), Systems & Storage (1), Elzar (9), Contact, Containers, GPU nodes, Jupyter, Login/Development nodes, MATLAB, UGE (Workload Manager), User Environment, and Workflow Tools. The main content area features a cartoon illustration of a scientist holding a brain. Text on the page describes Elzar as an HPC cluster introduced in 2020, consisting of 50 nodes (two head, two development, 46 compute) connected via dual 25 Gbps Ethernet networks, with a DDN GridScaler storage system. It also details the UGE workload management system and the hardware of the regular compute nodes.

The Essence of Elzar

Elzar is an institutionally shared high performance computing (HPC) cluster introduced in 2020. The cluster is intended to support the full spectrum of scientific computing at CSHL.

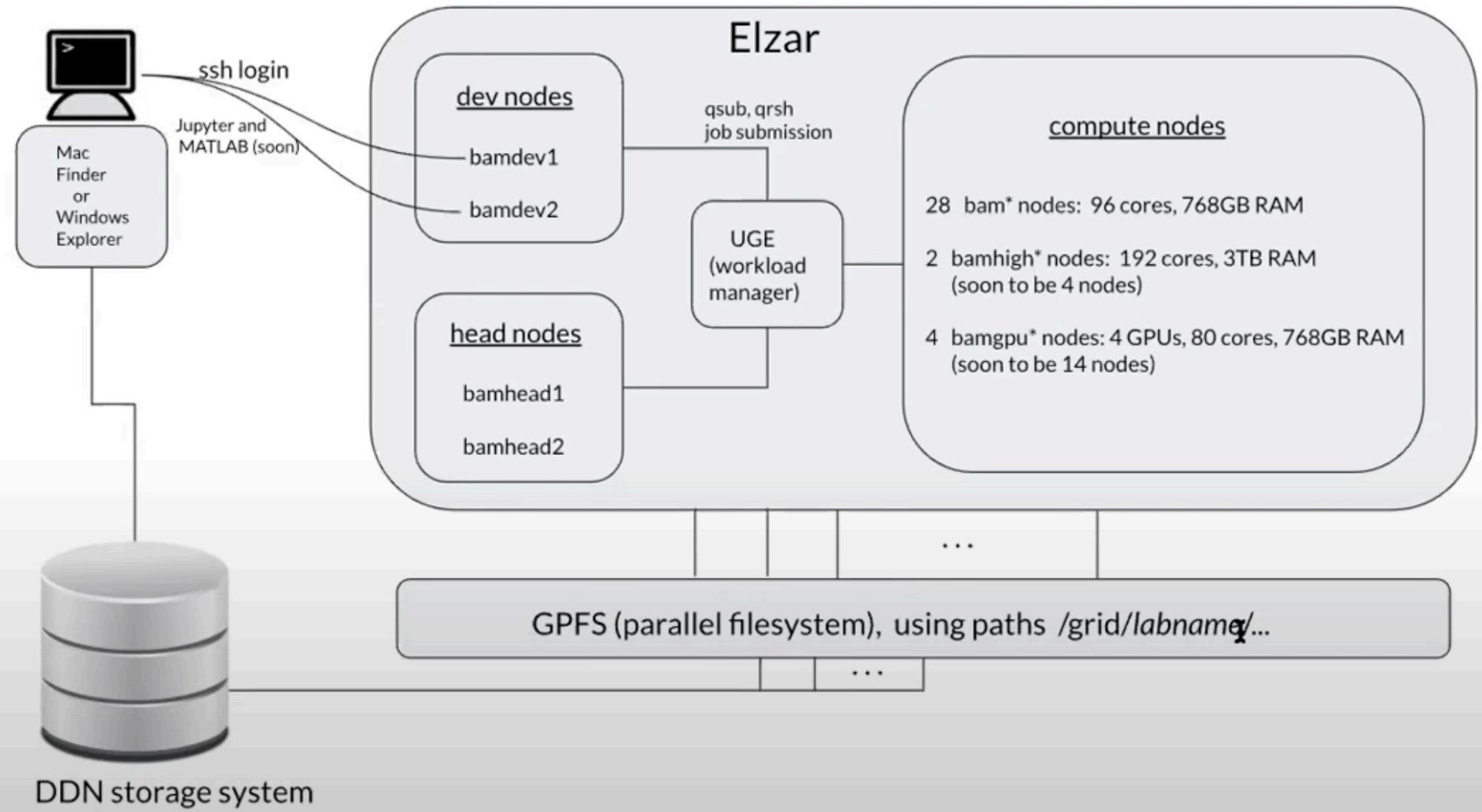
Elzar is a 2528 physical-core Dell PowerEdge system comprised of Intel Xeon Gold/Platinum (Cascade Lake-SP) processors. The cluster consists of 50 nodes: two head/management nodes, two development/login nodes, and 46 compute nodes. The nodes are connected via dual 25 Gbps Ethernet (GigE) networks. A DDN GridScaler storage system is accessed over Ethernet via a high performance parallel filesystem (GPFS).

The two development nodes provide login access to users and are meant for interactive development work, as well as for submitting production jobs to the workload manager (UGE) to run on the compute nodes. The cluster is administered by the pair of head nodes running UGE, which schedules the user jobs to run via a "fair share" resource management policy that equitably shares the processor and memory resources of the cluster. The two head nodes are configured for failover protection (high availability), which ensures that user job submission and execution is uninterrupted if one of the head nodes becomes unavailable.

The compute nodes that run user jobs submitted to UGE consist of 28 regular nodes, 4 high memory nodes, and 14 GPU nodes. The regular nodes have dual Xeon 6252 processors running at 2.1GHz with 24 cores per processor, and 768GB of memory. The high memory nodes have

If you use Elzar *at all* for your work, please cite “NIH Grant S10OD028632-01”

Architecture of Elzar



Elzar tutorial on YouTube

https://www.youtube.com/embed/D3wfhM_cQPY

The screenshot shows a Mac OS X desktop environment. In the foreground, a terminal window is open with a root shell session on a CSHL cluster node named 'bamhead1'. The user is demonstrating how to use the 'grid' command to access storage. The terminal output includes commands like 'module avail', 'module load <module>', and various 'grid' and 'findmnt' commands. In the background, a web browser window displays the CSHL Intranet homepage, specifically the 'Elzar - Google Slides' section. A video call is visible in the top right corner, showing Todd Heywood. The desktop dock at the bottom contains icons for various applications including Mail, Calendar, and Finder.

root@bamhead1:~\$ cutter:~ heywood\$ cutter:~ heywood\$ ssh he Welcome to Elzar See <http://intranet.cshl.edu> Use the following command "module avail" "module load <module>"

Last login: Thu Oct 22 1 [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ pma /grid/it/home/heywood [heywood@bamdev1 ~]\$ ls backup bnb bnlmod.sh [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ [heywood@bamdev1 ~]\$ fin |-/grid/it/home |-/grid/it/home_nlsas [heywood@bamdev1 ~]\$ ls total 197 drwxr-xr-x 4 heywood i drwxr-xr-x. 14 heywood i -rwxr-xr-x 1 heywood w drwxrwxr-x 3 heywood drwxr-xr-x 2 heywood i drwxr-xr-x 2 heywood i -rwxr-xr-x 1 heywood i drwxr-xr-x 3 heywood i drwxr-xr-x 2 heywood i -rw-r--r-- 1 root r -rwxr-xr-x 1 root r drwxr-xr-x 2 heywood i -rwxr-xr-x 1 heywood i drwxr-xr-x 4 heywood i drwxr-xr-x 2 heywood i [heywood@bamdev1 ~]\$

CSH Intranet

HOME GENERAL INFO ADMINISTRATION EDUCATION RESEARCH REQUESTS

| grid:/ware/data_nlease_norepl | grid[/ware/nlease_norepl]/data/data |

The simplified paths on the left map to the "real" paths on the right (where you should substitute "/mnt/grid/" for "grid["). Note that the "hpc" and "nlsas" labels used to distinguish between performance classes of storage, a distinction that no longer applies (the labels are just names now). The ".norepl" label means that the data at the path is not replicated to another physical location to protect against loss in case of a disaster like fire at the main location.

You may access your storage directly from your Mac or Windows machine using volume or share names. On Macs, open Finder and then select "connect to server" under the Go menu. Then enter "smb://grid-hs". You will see a list of volumes corresponding to the right side of the findmnt output described above, except showing all labs, from which you can then select your volume.

On Windows, open Explorer and click in the address bar, or type **ctrl+L**. Then enter "\grid-hs". Then you will see a list of shares for all labs corresponding to the right side of the findmnt output described above, which you can then select.

We plan on adding volume/share names matching the simplified paths on the left side of the findmnt output.

Slack workspace

An Elzar Users workspace has been set up on Slack. This may or may not turn out to be useful for users to share methods, ask questions, and get answers in a group environment. We'll see! If you use a cshl.edu email address, you can sign up at [this link](#). If you want to sign up without a cshl.edu email address, contact [Todd Heywood](#).

HELPFUL LINKS

- Campus Map
- Emergency Information
- Employee Self Service
- Faculty & Staff Directory (FACES)
- Shuttle Schedule

MENUS

- Blackford Hall
- Blackford Bar
- Hillside Cafe
- Genome Center
- Blackford Dinner

EXTERNAL LINKS

- CSHL External Website
- Labdish Blog
- Newsletter Signup
- Harbor Transcript
- Faculty & Staff

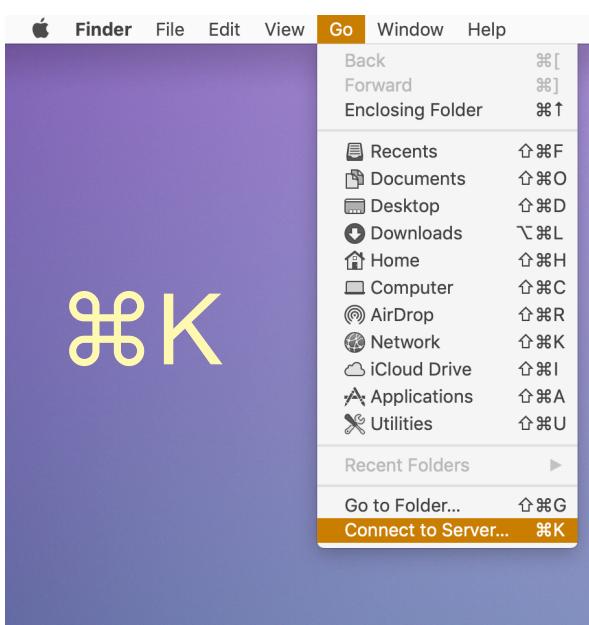
CONNECT WITH CSHL

- [Facebook](#)
- [Twitter](#)
- [YouTube](#)
- [LinkedIn](#)
- [Instagram](#)
- [Email](#)
- [RSS](#)

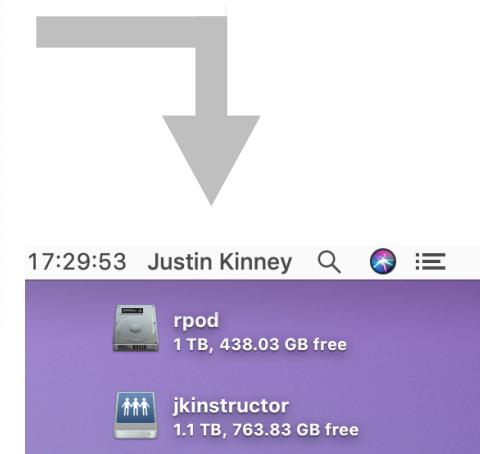
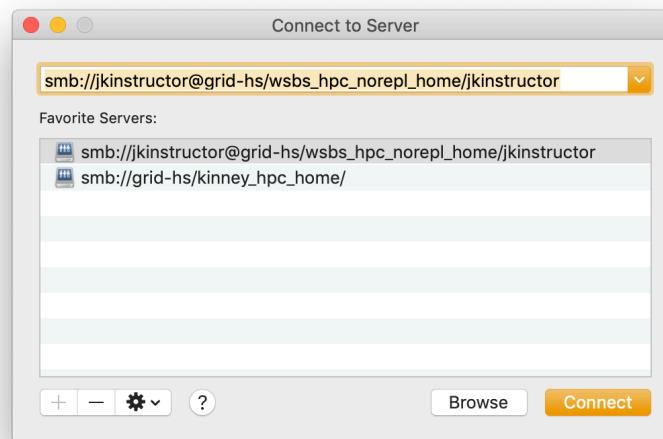
WHISTLEBLOWER POLICY

[View Whistleblower Policy](#)

The Elzar disk can be mounted using smb

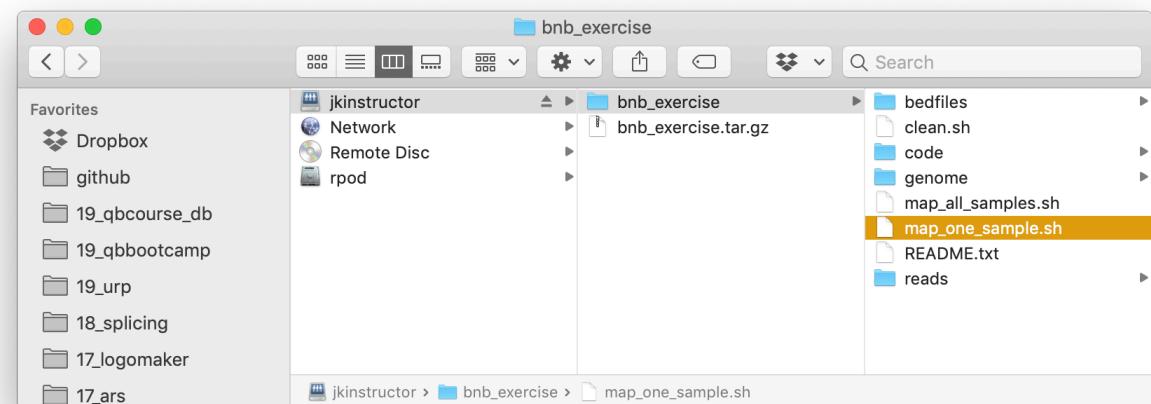


smb://you@grid-hs/wsbs_hpc_norepl_home/you
or
smb://you@grid-hs/yourlab_hpc_home/you



```
map_one_sample.sh
1 #!/usr/bin/env bash
2
3 # map_one_sample.sh
4 #
5 # Creates pileup files in .bed format for 4 Illumina
6 # samples
7 echo "Running single_process.sh..."
8
9 # Assign variables governing mapping
10 batch="A1"
11 read_length="31"
12
13 # Create output directories
14 echo "Setting up working area..."
15 ./clean.sh
16 mkdir mappings pileups
17
18 # Create bwa index for genome
19 echo "Creating index for genome..."
20 bwa index genome/genome.fasta
```

A screenshot of a terminal window titled 'map_one_sample.sh'. It displays a shell script with numerous lines of code, primarily in green and blue. At the bottom left, it says '110 Words, Line 1, Column 1' and 'Tab Size: 4'.



```
Desktop — jkinstructor@bamdev2:~ — ssh jkinstructor@bamdev2 — 106x20

(base) jkinney@dobbs:~/Desktop$ ssh jkinstructor@bamdev2
[jkinstructor@bamdev2's password:
Permission denied, please try again.
[jkinstructor@bamdev2's password:
Permission denied, please try again.
[jkinstructor@bamdev2's password:
Welcome to Elzar
See https://intranet.cshl.edu/info-technology/it-groups/systems-and-storage-group/elzar/
Use the following commands to adjust your environment:

'module avail'           - show available modules
'module load <module>'  - adds a module to your environment for this session

-----
Last failed login: Tue Sep  5 07:10:58 EDT 2023 from 143.48.116.1 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Wed Aug 24 16:32:35 2022 from 143.48.116.10
[[jkinstructor@bamdev2 ~]$ ls
elzar_exercise  elzar_exercise.tar.gz
[jkinstructor@bamdev2 ~]$
```



jkinney — jkinstructor@bamdev1:~ — ssh jkinstructor@bamdev1 — 93x33

```
[[jkinstructor@bamdev1 ~]$ ls
elzar_exercise.tar.gz
[[jkinstructor@bamdev1 ~]$ tar -zxf elzar_exercise.tar.gz
elzar_exercise/
elzar_exercise/bedfiles/
elzar_exercise/elzar_modules.sh
elzar_exercise/code/
elzar_exercise/map_all_samples.sh
elzar_exercise/reads/
elzar_exercise/clean.sh
elzar_exercise/README.txt
elzar_exercise/map_one_sample.sh
elzar_exercise/genome/
elzar_exercise/genome/genome.fasta
elzar_exercise/reads/D1.fastq
elzar_exercise/reads/B1.fastq
elzar_exercise/reads/C1.fastq
elzar_exercise/reads/A1.fastq
elzar_exercise/code/pileup2bedfile.py
[[jkinstructor@bamdev1 ~]$ ls
elzar_exercise  elzar_exercise.tar.gz
[jkinstructor@bamdev1 ~]$
```

```
jkinney — jkinstructor@bamdev1:~/elzar_exercise — ssh jkinstructor@bamdev1 — 93x33
[[jkinstructor@bamdev1 ~]$ cd elzar_exercise/
[[jkinstructor@bamdev1 elzar_exercise]$ ls
bedfiles  code          genome          map_one_sample.sh  reads
clean.sh  elzar_modules.sh  map_all_samples.sh  README.txt
[[jkinstructor@bamdev1 elzar_exercise]$ cat README.txt
### Scripts ###

source elzar_modules.sh -- Load all modules required by the tools.
./clean.sh -- Cleans out all analysis files from area
./map_one_sample.sh -- Maps reads in one file to the yeast genome and creates bed pileups
./map_all_samples.sh -- Maps reads in 4 files to the yeast genome and creates bed pileups

### Input files ###

./genome/
    genome.fastq    # Yeast genome

./reads/
    A1.fastq  B1.fastq  C1.fastq  D1.fastq    # 4 read files

### Dependencies ###

./code/pileup2bedfile.py -- Creates bed file from samtools pileup file

[[jkinstructor@bamdev1 elzar_exercise]$ source elzar_modules.sh
[jkinstructor@bamdev1 elzar_exercise]$ ]
```



jkinney — jkinstructor@bamdev1:~/elzar_exercise — ssh jkinstructor@bamdev1 — 93x33

```
[[jkinstructor@bamdev1 elzar_exercise]$ ./map_one_sample.sh
Running single_process.sh...
Setting up working area...
Creating index for genome...
[bwa_index] Pack FASTA... 0.06 sec
[bwa_index] Construct BWT for the packed sequence...
[bwa_index] 2.79 seconds elapse.
[bwa_index] Update BWT... 0.06 sec
[bwa_index] Pack forward-only FASTA... 0.04 sec
[bwa_index] Construct SA from BWT and Occ... 1.11 sec
[main] Version: 0.7.17-r1188
[main] CMD: bwa index genome/genome.fasta
[main] Real time: 4.347 sec; CPU: 4.074 sec
Mapping reads to genome...
[bwa_aln] 17bp reads: max_diff = 2
[bwa_aln] 38bp reads: max_diff = 3
[bwa_aln] 64bp reads: max_diff = 4
[bwa_aln] 93bp reads: max_diff = 5
[bwa_aln] 124bp reads: max_diff = 6
[bwa_aln] 157bp reads: max_diff = 7
[bwa_aln] 190bp reads: max_diff = 8
[bwa_aln] 225bp reads: max_diff = 9
[bwa_aln_core] calculate SA coordinate... ]
```



JKinney — jkinstructor@bamdev1:~/elzar_exercise — ssh jkinstructor@bamdev1 — 93x30

```
[jkinstructor@bamdev1 elzar_exercise]$ ./map_all_samples.sh
Running single_process.sh...
Setting up working area...
Creating index for genome...
[bwa_index] Pack FASTA... 0.07 sec
[bwa_index] Construct BWT for the packed sequence...
[bwa_index] 2.96 seconds elapse.
[bwa_index] Update BWT... 0.05 sec
[bwa_index] Pack forward-only FASTA... 0.04 sec
[bwa_index] Construct SA from BWT and Occ... 1.06 sec
[main] Version: 0.7.17-r1188
[main] CMD: bwa index genome/genome.fasta
[main] Real time: 4.479 sec; CPU: 4.185 sec
Submitting scripts/map_A1.sh to cluster...
```

```
Your job 4708163 ("map_A1.sh") has been submitted
Submitting scripts/map_B1.sh to cluster...
```

```
Your job 4708164 ("map_B1.sh") has been submitted
Submitting scripts/map_C1.sh to cluster...
```

```
Your job 4708165 ("map_C1.sh") has been submitted
Submitting scripts/map_D1.sh to cluster...
```

```
Your job 4708166 ("map_D1.sh") has been submitted
Waiting for scripts to finish ...
-> 4 jobs left. Waiting 10s...
```

jkinney — jkinstructor@bamdev1:~/elzar_exercise — ssh jkinstructor@bamdev1 — 93x16

```
[[jkinstructor@bamdev1 elzar_exercise]$ ls
bedfiles          genome          map_B1.sh.e4708164  map_D1.sh.e4708166  reads
clean.sh          map_A1.sh.e4708163  map_B1.sh.o4708164  map_D1.sh.o4708166
code              map_A1.sh.o4708163  map_C1.sh.e4708165  map_one_sample.sh
elzar_modules.sh  map_all_samples.sh  map_C1.sh.o4708165  README.txt
[[jkinstructor@bamdev1 elzar_exercise]$ ls -lah bedfiles/
total 35M
drwxr-s--- 2 jkinstructor wsbs students 4.0K Aug 30 07:05 .
drwxr-sr-x 6 jkinstructor wsbs students 32K Aug 30 07:05 ..
-rw-r--r-- 1 jkinstructor wsbs students 8.7M Aug 30 07:04 A1.pileup.bed
-rw-r--r-- 1 jkinstructor wsbs students 8.7M Aug 30 07:04 B1.pileup.bed
-rw-r--r-- 1 jkinstructor wsbs students 8.7M Aug 30 07:04 C1.pileup.bed
-rw-r--r-- 1 jkinstructor wsbs students 8.7M Aug 30 07:04 D1.pileup.bed
[jkinstructor@bamdev1 elzar_exercise]$
```

To do this morning:

1. Copy **elzar_exercises.tar.gz** from **22e_qbbootcamp/** to your home directory on Elzar
2. Map one sample of reads to genome using **map_one_sample.sh**
3. Submit four mapping jobs to cluster using **map_all_samples.sh**
4. Copy .bed files to local machine
5. This afternoon: Visualize replication profiles using Python.