

# Discovering Origins of Replication

Michael Schatz & Justin Kinney

Sept 3, 2013

QB Bootcamp Lecture 5



# Outline

Part 1: Overview & Fundamentals

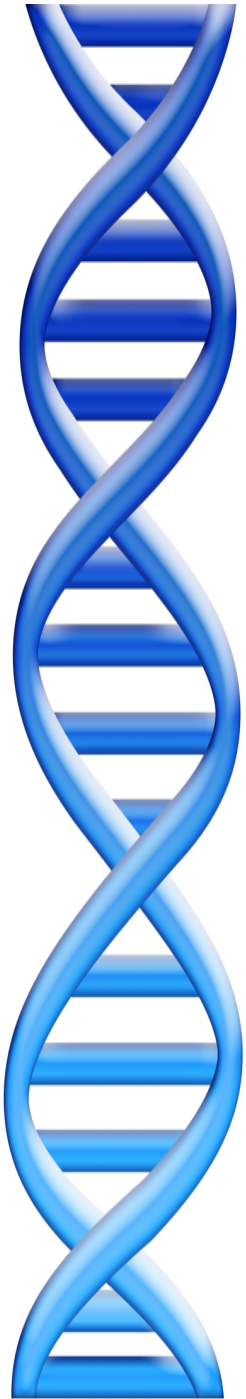
Part 2: Sequence Analysis Theory

Part 3: Genomic Resources

Part 4: Unix Scripting

**Part 5: Example Analysis**

- Background on tracking DNA replication with next-gen sequencing
- Walk-through of analysis steps
- Visualization of discovered replication sites



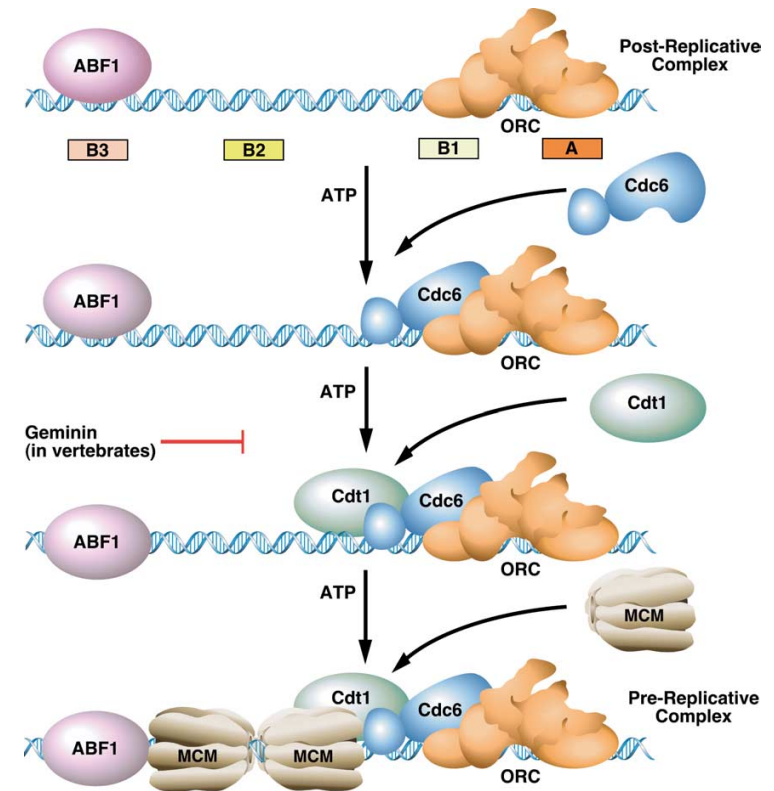
# ~300 separate loci direct DNA replication initiation in *Saccharomyces cerevisiae*

ARS: autonomously replicating sequence

G1



S



The Stillman lab is interested, in part, in the signaling mechanisms governing pre-RC firing -> genome-wide replication tracking

# Tracking replication with EdU pulldown + sequencing

DNA of cells arrested in G1 with  $\alpha$ -factor

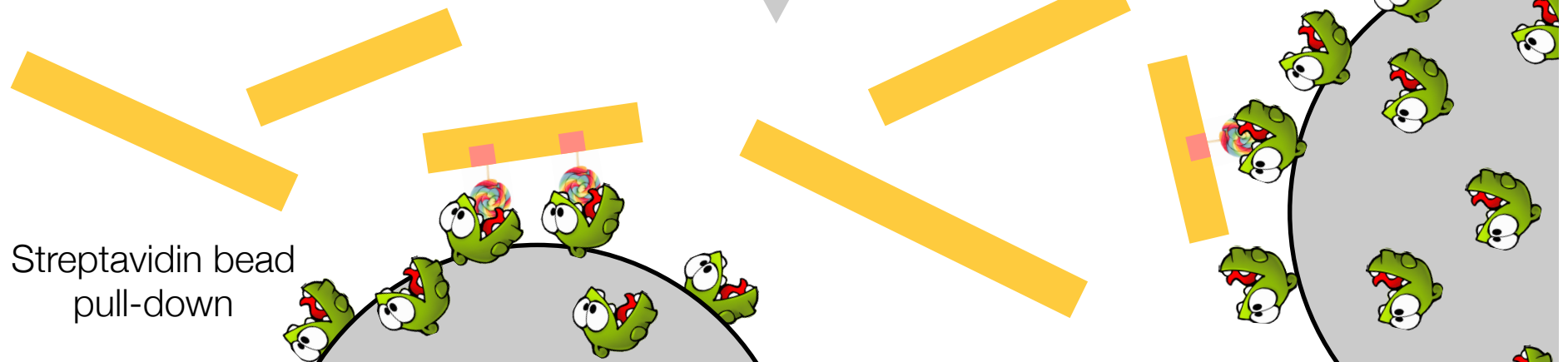


Release cells into S-phase  
EdU incorporation during replication

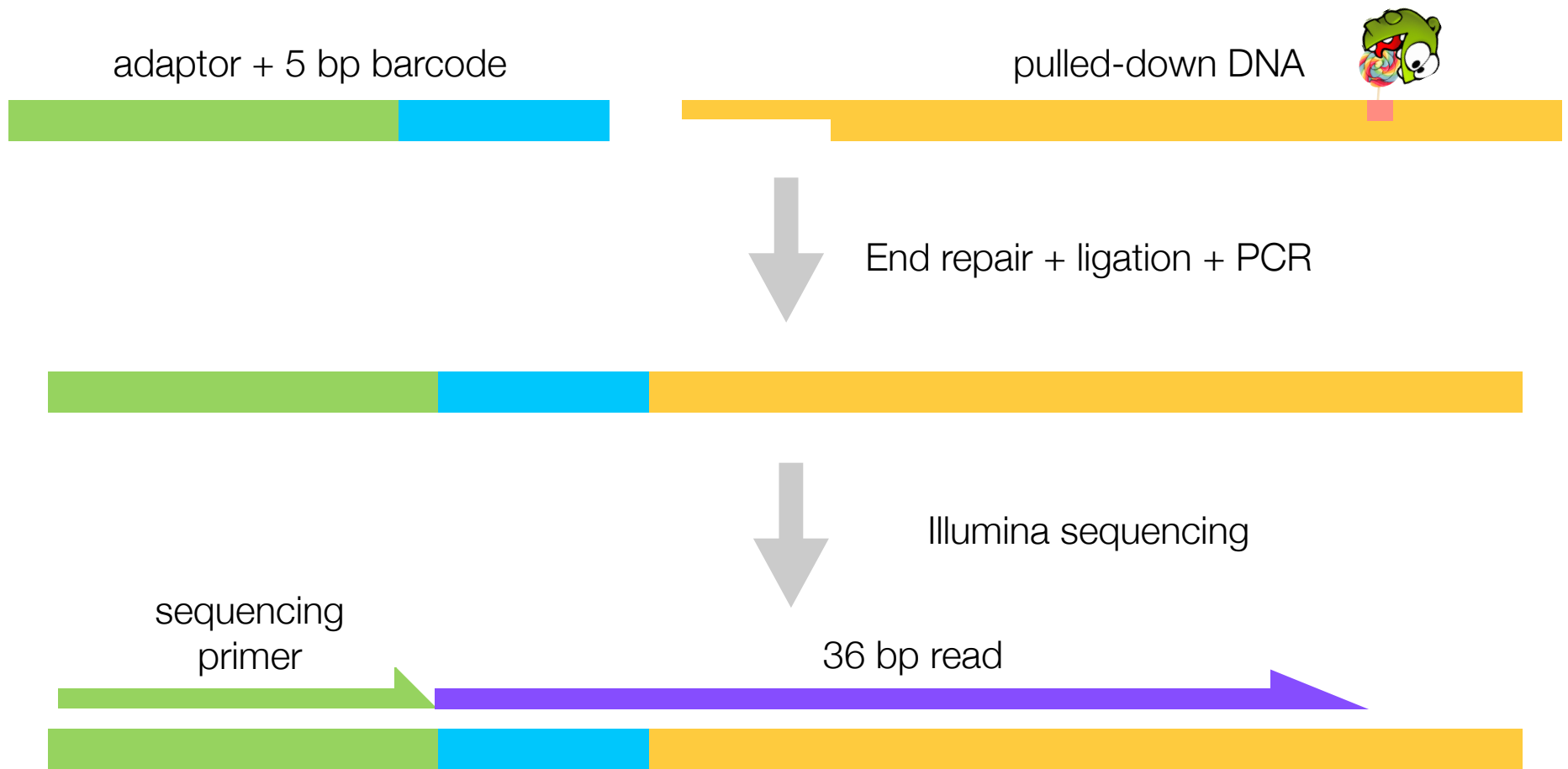


Click-iT linking of EdU to biotin

DNA sonication



# Barcoding samples for sequencing



~15 M reads for 14 barcoded samples

**Thanks Yi-Jun!**

# What we will do

- Today
  - Map reads to the yeast genome
  - Compute “replication profiles”: # of reads covering each genomic position
  - View these data using the UCSC genome browser; compare to known ARSs
- Tomorrow
  - Python tutorial (cont)
  - Load replication profiles into Python
  - Smooth and plot replication profiles

# Analysis Pipeline



- No single application available that will let us analyze these data
  - Just 4 steps to go from raw observations to biological discovery
- Each step requires selection, tuning, and debugging
  - Analogous to a wetlab protocol for running an experiment
- The components of the pipeline can be used in many other assays
  - Reads => Comparative Genomics, Transcriptome Analysis, de novo sequencing, Protein binding sites, Chromatin regulation...
  - Alignment => Forms the basis for almost every assay
  - SAMTools => Filtering, selection, interpretation of alignments

# Analysis Pipeline



- Get the files (curl dash Capital-O)

```
$ curl -O http://schatzlab.cshl.edu/data/challenges/replication_exercise.tgz
```

- Unpack the files

```
$ tar xzvf replication_exercise.tgz
```

- Check out the files

```
$ cd replication_exercise/  
$ ls -R  
$ less *.txt  
$ less reads/A1.fastq
```

[What is the secret phrase?]



# Analysis Pipeline



- Check out the analysis script

```
$ cat course_pipeline.sh
```

- We have already done the first steps to partition reads into batches

```
# Quality filter reads
# fastq_quality_filter -q 10 -p 90 -i /data/kinney/data/illumina_sequencing/
11.01.24_sheu_edu/reads.fastq -o reads/reads_qual.fastq

# Split reads by batch
# cat reads/reads_qual.fastq | fastx_barcode_splitter.pl --bcfile /data/
kinney/data/illumina_sequencing/11.01.24_sheu_edu/barcodes.txt --prefix reads/
tmp1_ --suffix .fastq --mismatches 0 -bol
```

- You can embed comments into scripts with '#'

# Analysis Pipeline



- Now that the reads are prepared, next step is to align

```
# Create bwa index for genome  
# bwa index genome/genome.fasta
```

```
# Align reads using bwa  
bwa aln genome/genome.fasta reads/A1.fastq > mappings/A1.sai  
bwa samse genome/genome.fasta mappings/A1.sai reads/A1.fastq > mappings/A1.sam
```

- BWA (Li & Durbin, 2009) is one of the most popular tools for aligning short reads to a reference genome. It is used in almost every sequencing assay that start from short reads. It takes a few steps to run because it uses the special BWT index of the genome for making the alignments fast.

# Analysis Pipeline



- Now that the reads are aligned, need to transform and sort them

```
# Create pileup using samtools
samtools view -bS mappings/A1.sam > mappings/A1.bam
samtools sort mappings/A1.bam mappings/A1.sorted
samtools index mappings/A1.sorted.bam
samtools pileup -c -f genome/genome.fasta mappings/A1.sorted.bam > pileups/A1.pileup
```

- The pileup file encodes how many reads align to each position in the genome

```
$ less pileups/A1.pileup
```

- Run a quick command to find positions with deep coverage

```
$ awk '{if ($8>50){print}}' A1.pileup | less
```

[AWK is a really powerful, if arcane filter]

# Analysis Pipeline



- Now run a custom script to summarize the depth information

```
$ ./pileup2bedfile.py pileups/A1.pileup 31  
$ less pileups/A1.pileup.bed
```

- This file can then be loaded into the UCSC Genome Brower for inspection, and relate it to known annotations

See <http://genome.ucsc.edu/>

# Exercise & Homework

- Replication Analysis
  - Use Galaxy to identify any SNPs in AI, BI, CI, or DI files
  - Modify course\_pipeline.sh to analyze BI, CI, DI
  - Load the bed files into the UCSC genome browser
  - See if you can spot and interesting variations between the data sets
- Need more help?
  - <http://www.codecademy.com/tracks/python>
  - <http://docs.python.org/3/tutorial/index.html>
  - <http://www.ee.surrey.ac.uk/Teaching/Unix/>

# Questions?

<http://schatzlab.cshl.edu>