Combinatorial Pooling Enables Selective Sequencing of the Barley Gene Space

Stefano Lonardi

Department of Computer Science and Engineering

University of California, Riverside

Joint work with D. Duma (UCR), M. Alpert (UCR), F. Cordero (U. of Torino), M. Beccuti (U. of Torino), P. R. Bhat (UCR and Monsanto), Y. Wu (UCR and Google), G. Ciardo (UCR), B. Alsaihati (UCR), Y. Ma (UCR), S. Wanamaker (UCR), J. Resnik (UCR), and T. J. Close (UCR)

Combinatorial Pooling for Genomics

- *Resequencing*: structural variations (SNPs)
- *Screening*: protein-protein interaction, compound screening
- *Metagenomics*: identification of species in a sample
- This talk: *de novo* genome sequencing

de novo sequencing

- Current estimates: 8.7 million (±1.3 million SD) eukaryotic species on the planet [Mora *et al.*, *PLoS Biology* 2011]

- Genome sequence is available (at different level of completion) only for a few hundred eukaryotes

Barley genome (*H. vulgare*)

- Diploid
- Seven chromosomes
- Highly repetitive (>90%)
- Size is ≈5.3 Gb
  - ≈12x the size of rice
  - ≈36x the size of *arabidopsis*

Barley genome (*H. vulgare*)

- Genome too large/repetitive/expensive for whole shotgun sequencing
- BAC: an *E.coli* cell containing a ~100-150kb fragment of the barley genome
- Genes are not distributed evenly along the genome: they are clustered in gene-rich regions, thus a BAC carrying one gene is likely to carry several genes
- Strategy (*selective sequencing*)
  - Identify gene-enriched BACs
  - Build an overlap map (*physical*) for these BACs
  - Sequence only a non-redundant subset of them (*MTP*)

BAC-by-BAC vs. shotgun sequencing

- Pros
  - Can be selective (*i.e.*, gene enrichment)
  - Work can be distributed across several labs
  - Assembly can be carried out BAC-by-BAC (helps dealing with high repeat content)
- Cons
  - Need BAC overlap map (*physical map*)
  - *E. coli* contamination
  - Need to handle large number of individual samples

Barley BAC physical map

- Started from 6.64x genome equivalent BAC library for barley (313,344 BACs)
- Selected 83,831 gene-positive BAC [Madishetty *et al.*, *NAR* 2006], then fingerprinted using HICF (five restriction enzymes)
- A physical map of the BACs produced 6,579 *contigs* covering about a 1/3 of the barley genome [Bozdag *et al.*, *BMC Bioinfo* 2009]

Minimum Tiling Path (MTP)

- 15,820 BACs were identified as *minimal tiling path* (MTP) clones, for a total of ~1,700 Mb
  [Bozdag *et al.*, *Proc. WABI* 2008]

Next-Generation Sequencing

- NGS instruments have a fixed numbef of 'lanes' for DNA samples (e.g., Illumina has 8)
- Each lane produces a fixed amount of data (e.g. 10-100GB/lane on the Illumina)
- Allocating one BAC to each individual lane would be expensive and wasteful
- Need to pool many BACs on the same lane, but DNA barcoding does not scale to hundreds or thousands of samples

Combinatorial Pooling

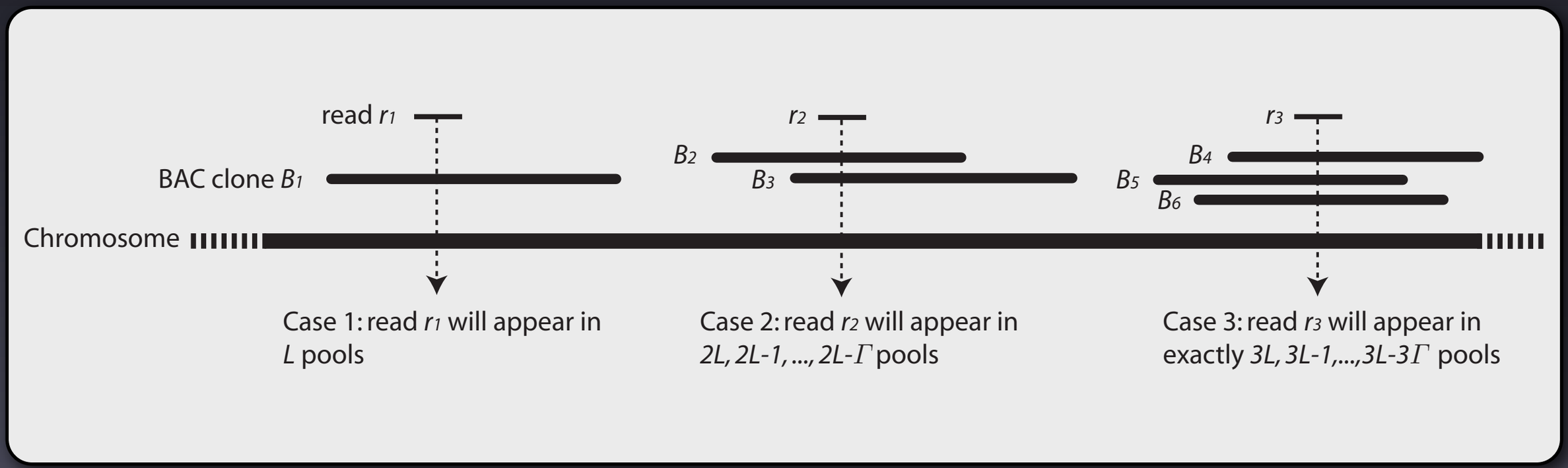
- *Idea:* Replicate each BAC in a set of pools according to a *combinatorial pooling* scheme so that the identity of a BAC is encoded in the pattern of pools (*signature*) where it is contained [by transitivity, corresponding sequenced reads will exhibit the same pool pattern]

Combinatorial Pooling

- A *shifted transversal* design is defined by (P,L,Γ,d) such that P is a prime, $P^{\Gamma+1} \geq N$ and $floor[(L\text{-}1)/\Gamma] \geq d$ [Thierry-Mieg, *BMC Bioinfo* 2006]

- Properties
  - Number of pools is PL
  - *Decodability* is d
  - A BAC is replicated in L pools
  - Each pool contains P^{Γ} BACs
  - Two BACs can share at most Γ pools

Need a 3-decodable design



Set L=7, Γ=2 → 3-decodable

Several 3-decodable 7-layer designs

P	BACs/pool (P^2)	Total BACs (P^3)	Total pools (7xP)	$\frac{\text{Total BACs}}{\text{Total pools}}$
7	49	343	49	7.0
11	121	1,331	77	17.3
13	169	2,197	91	24.1
17	289	4,913	119	41.3
19	361	6,859	133	51.6
23	529	12,167	161	75.6
29	841	24,389	196	124.4

Synthetic Data for Rice Genome

- We selected 2,197 MTP BACs from a real physical map of rice (~390Mb genome)
- BACs were pooled according to the ST design (*P=13, L=7, Γ=2, d=3)*
- 1M paired-end reads of 104 bases (with 1% sequencing error) were generated *in silico* for each pool, equivalent to 8x coverage for one BAC in one pool (56x overall)

Real Data for Barley Gene Space

- We divided the 15,820 barley MTP BACs in seven sets of 2,197 and pooled according to the ST design (*P=13, L=7, Γ=2, d=3)*
- Each set of 91 pools run on one Illumina flowcell: each of the seven available lanes was assigned 13 pools multiplexed via DNA-barcoding

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{01, 16, 34, 42, 53, 67, 84}

BAC
#0001

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{04, 18, 33, 49, 53, 71, 90}

BAC
#0002

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{09, 21, 04, 49, 65, 78, 88}

BAC
#0003

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{03, 21, 31, 46, 53, 78, 82}

BAC
#0004

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{07, 23, 39, 42, 58, 74, 90}

BAC
#0005

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{05, 17, 27, 48, 54, 71, 86}

BAC
#0006

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{13, 17, 37, 47, 60, 76, 82}

BAC
#0007

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{05, 25, 39, 47, 62, 71, 87}

BAC
#0008

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

BAC signature
{01, 16, 28, 50, 56, 72, 85}

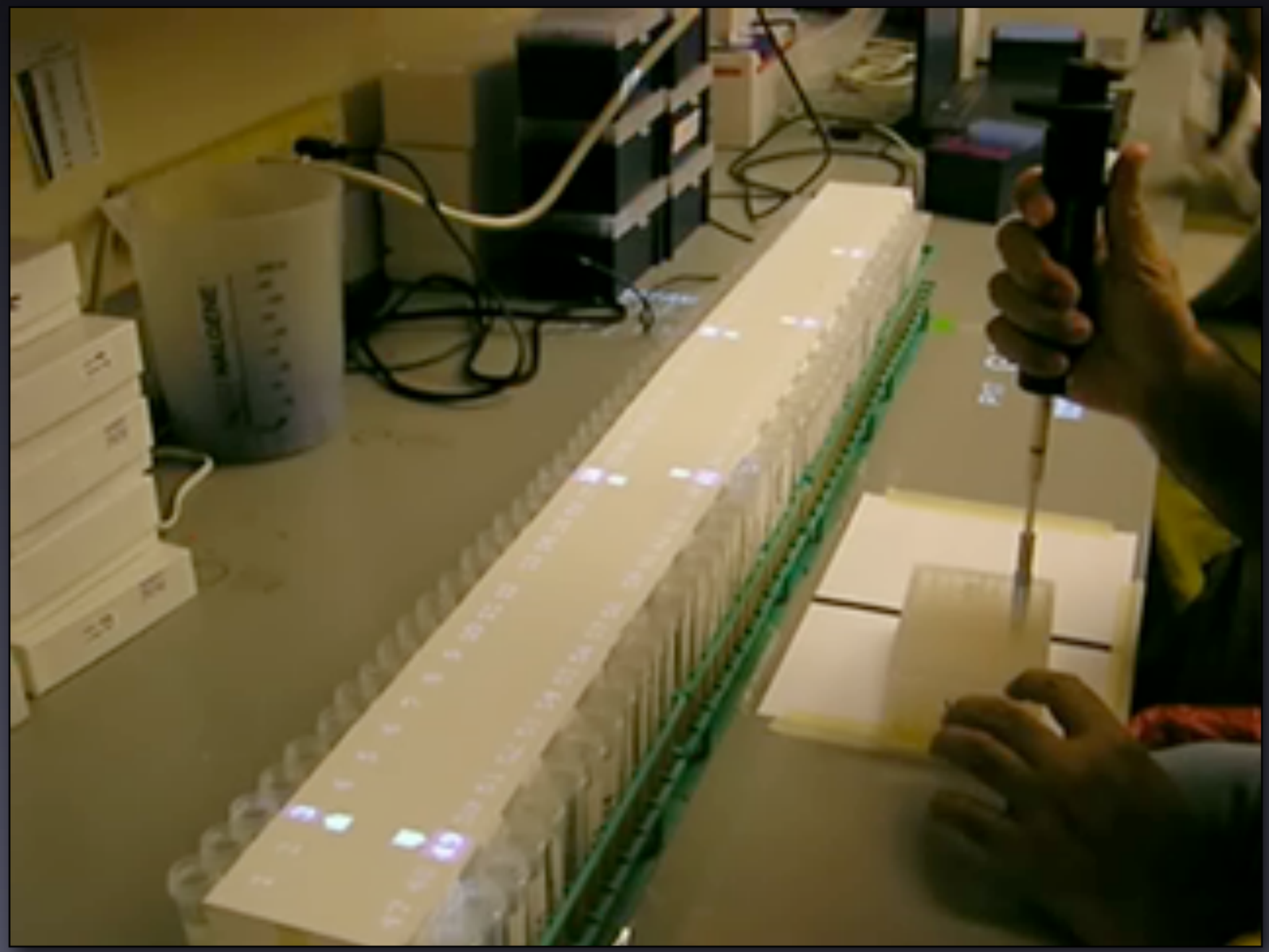
BAC
#0009

- 2197 BACs
- 91 pools: 7 layers, 13 pools per layer
- 169 BACs per pool
- Each BAC in 7 pools, one per layer

Layer													
1	01	02	03	04	05	06	07	08	09	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25	26
3	27	28	29	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49	50	51	52
5	53	54	55	56	57	58	59	60	61	62	63	64	65
6	66	67	68	69	70	71	72	73	74	75	76	77	78
7	79	80	81	82	83	84	85	86	87	88	89	90	91

... and so on for all 2,197 BACs ...

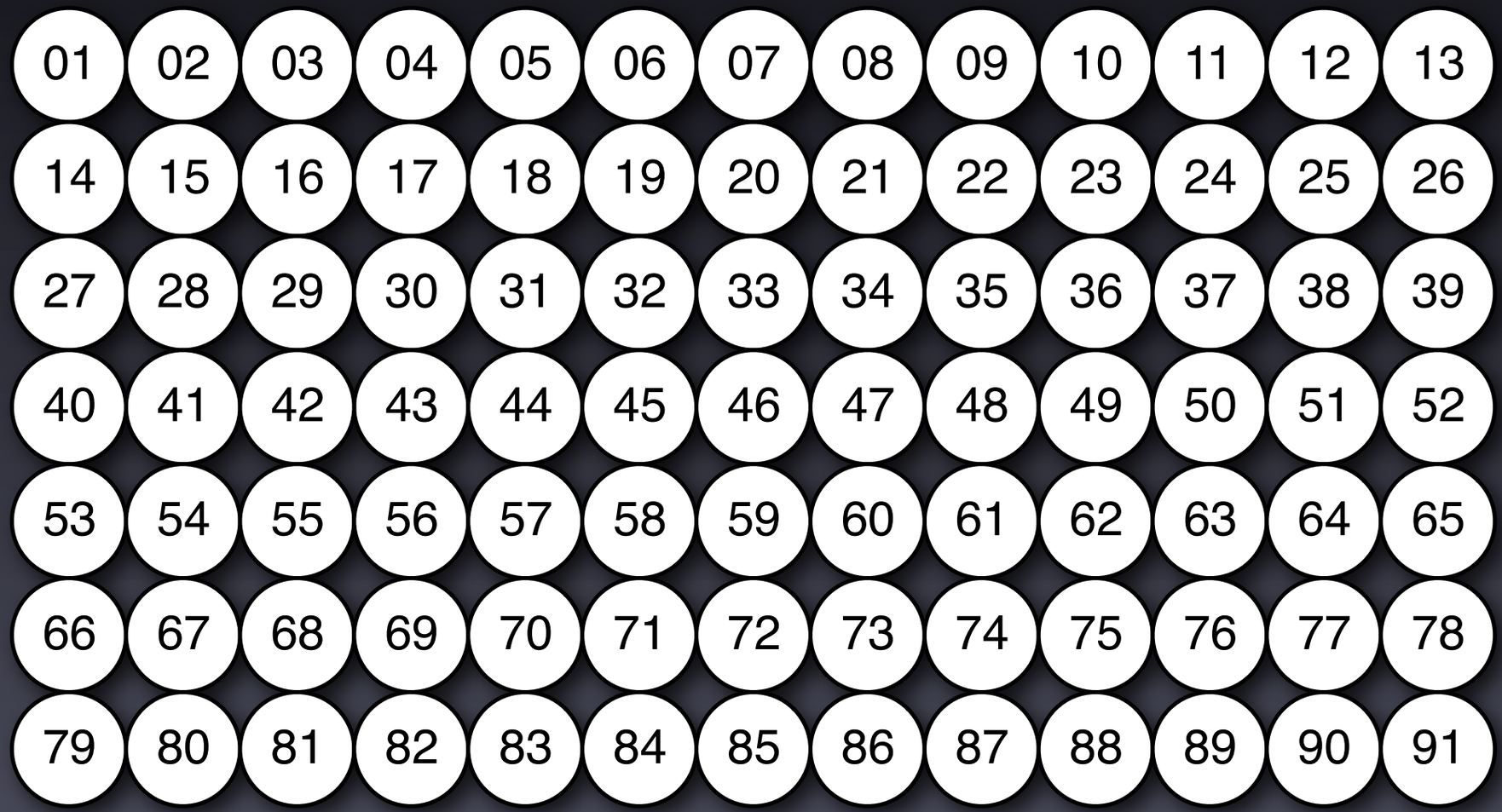
Pooling BAC clones



Real Data for Barley Gene Space

- We obtained an average of ~12.4M reads per pool with an average length of 94 bases
- After "cleaning" we ended up with an average of ~5.5M reads per pool, with an average length of 88 bases
- As a result, the average sequencing depth for a BAC was ~157x (before deconvolution)

Computing read signatures



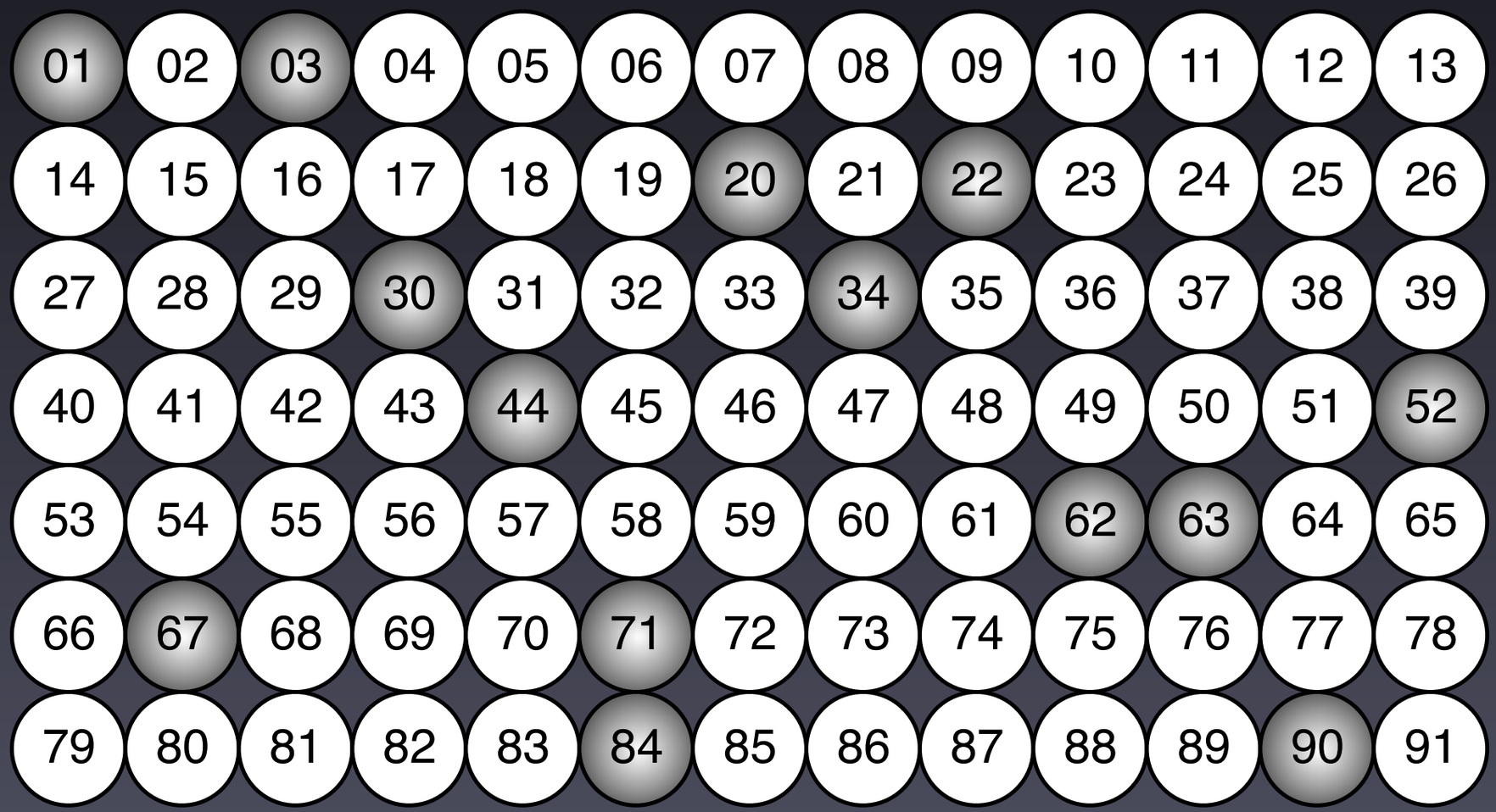
Which pools has an occurrence of *r*, say *r*=TACCATA... ?

What does it mean for *r* to occur in a pool *j*?

Occurrence of a read in a pool

- We cannot expect a full-length perfect match between read *r* and another read in pool *j*
- Due to sequencing errors, we have to allow for a limited number mismatches
- Need to allow for prefix-suffix overlap

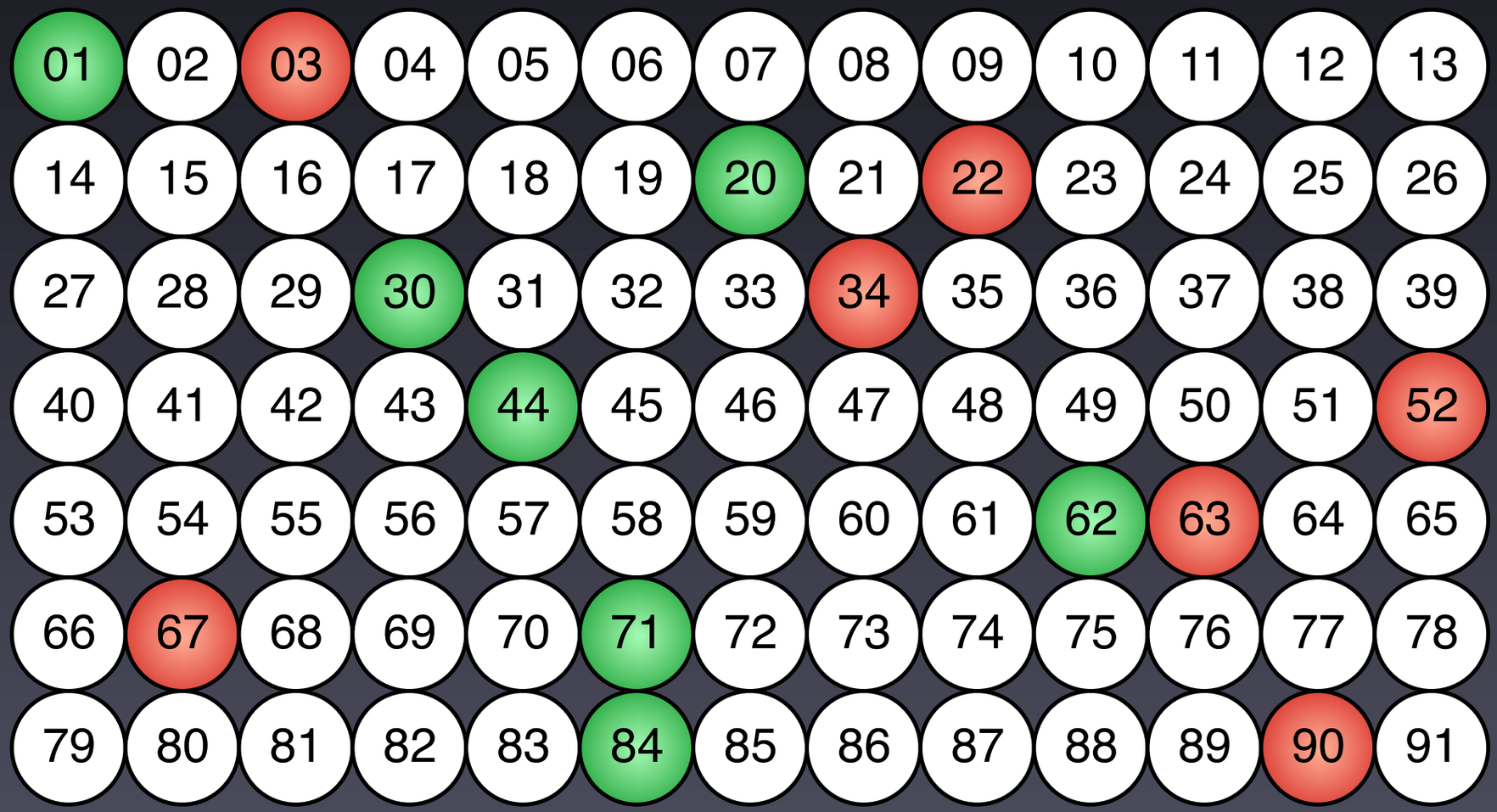
Decoding read signatures



Read signature

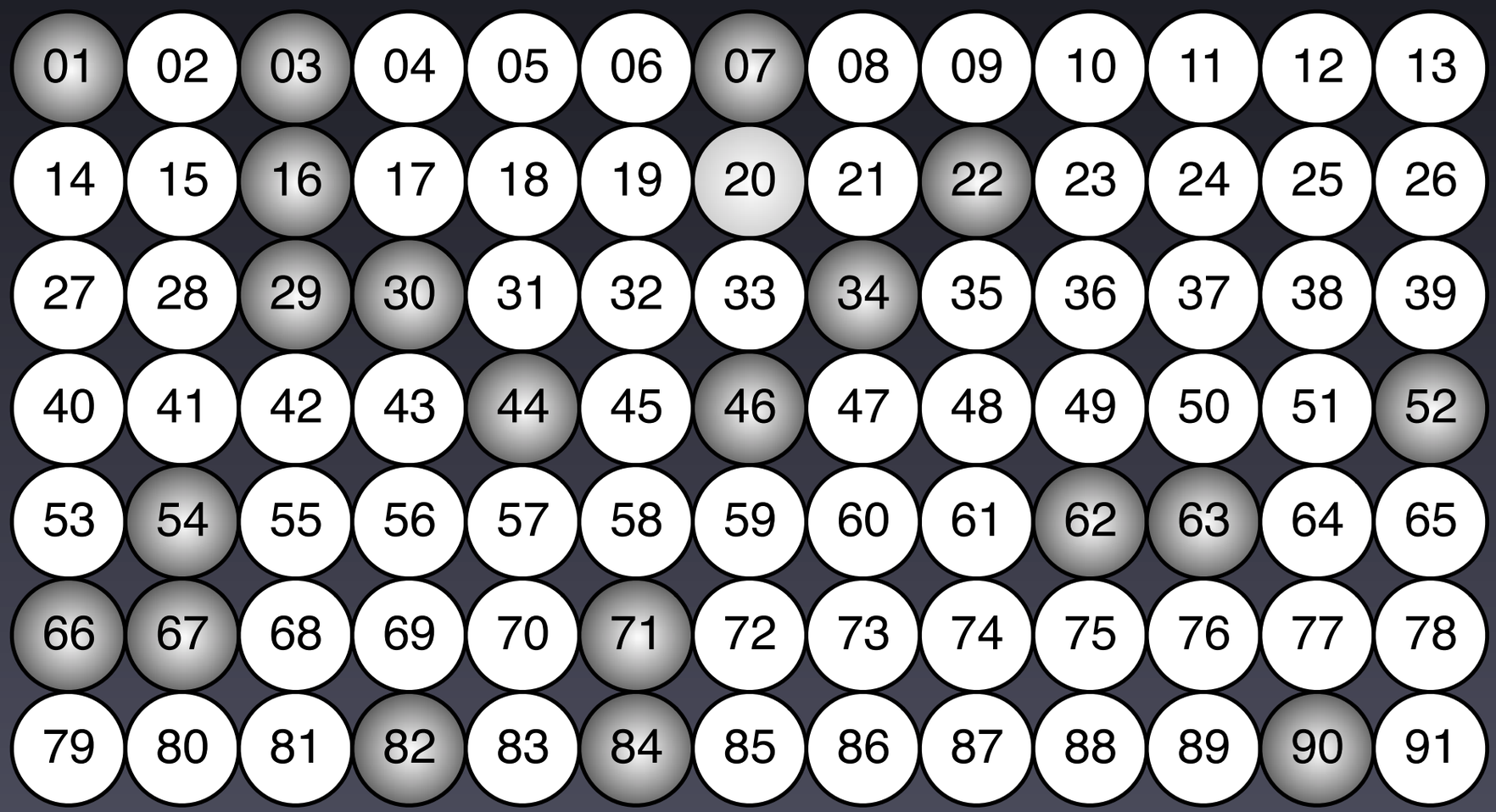
{01, 03, 20, 22, 30, 34, 44, 52, 62, 63, 67, 71, 84, 90}

Decoding read signatures



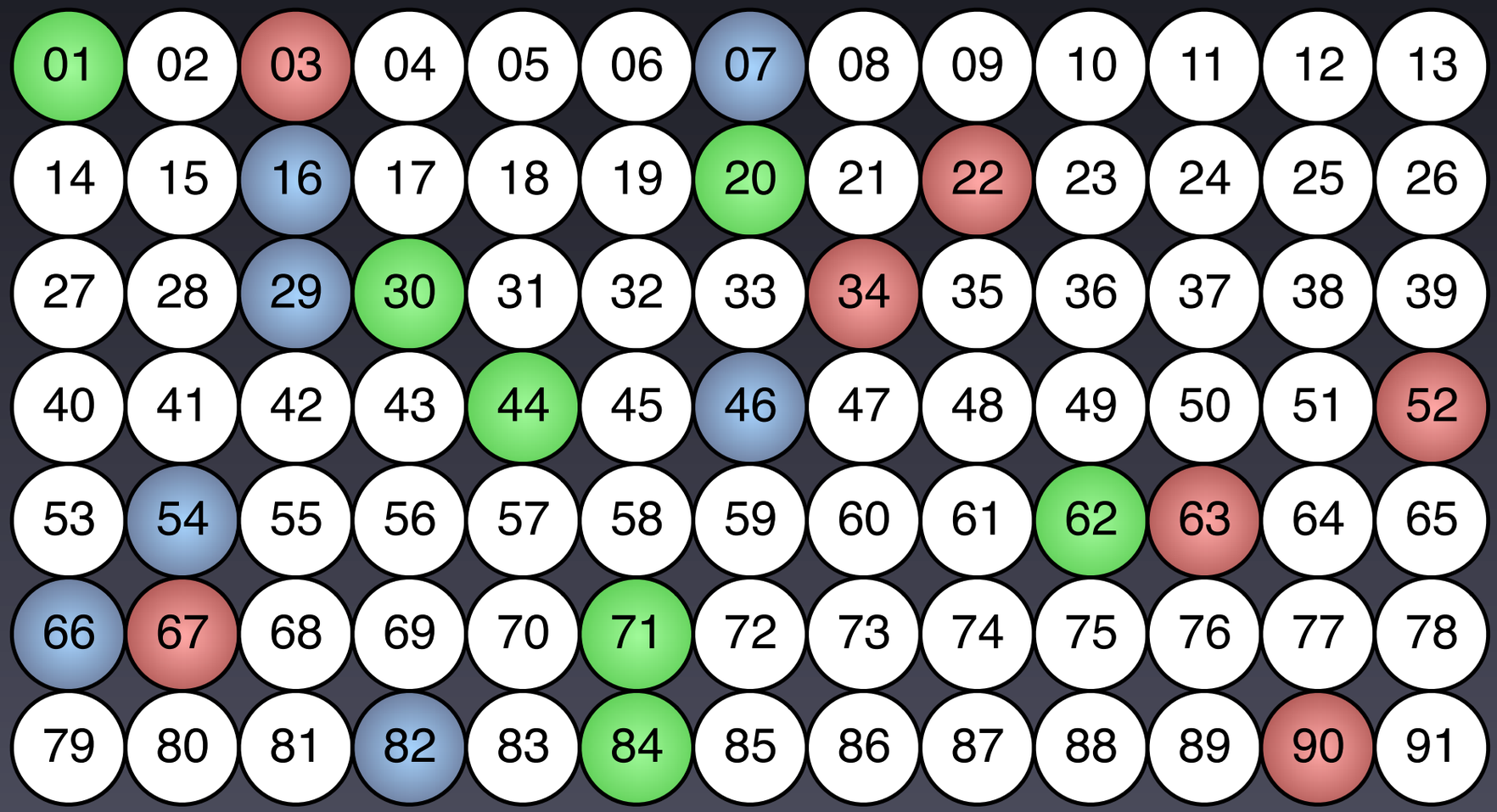
BAC signatures	BAC
{03, 22, 34, 52, 63, 67, 90}	#0296
{01, 20, 30, 44, 62, 71, 84}	#1179

Decoding read signatures



Read signature {01, 03, 07, 16, 20, 22, 29, 30, 34, 44, 46, 52, 54, 62, 63, 66, 67, 71, 82, 84, 90}

Decoding read signatures



{03, 22, 34, 52, 63, 67, 90} #0296

{01, 20, 30, 44, 62, 71, 84} #1179

{07, 16, 29, 46, 54, 66, 82} #1861

Deconvolution problem

- *Input:* Given a set of 91 pools of reads, and the signatures of 2,197 BACs
- *Output:* An assignment of each read to 1, 2 or 3 BACs
- *Challenge:* The total number of input reads is in the hundreds of millions; need an accurate time- and memory-efficient method to compute the signature of all the reads

Initial Attempts

- Implemented a prefix-suffix approximate overlap method based on hash-tables
- Tested a recently published prefix-suffix approximate overlap method based on the FM-index [Välimäki *et al.*, *Proc CPM* 2010]
- Tested the experimental short-read assembler SGA, which also uses the FM-index [Simpson *et al.*, *Genome Res.* 2012]
- Idea: use the shared k-mer content, no need to compute actual overlaps

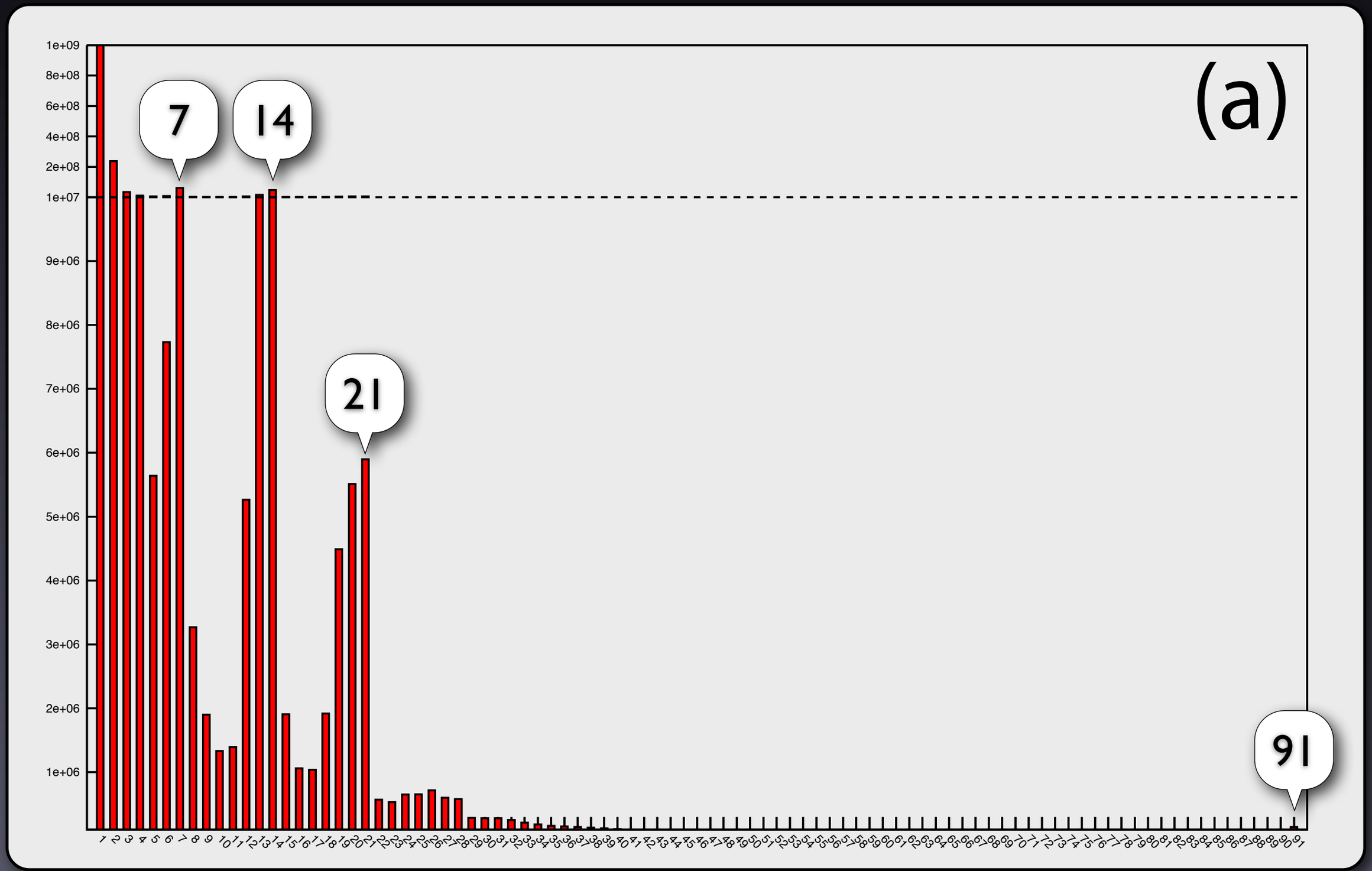
HashFilter's k-mer based strategy

1. *Preprocessing*: for each each distinct k-mer *w*, compute the number of exact occurrences of *w* or w^{rc} in each pool (*frequency vector*)
2. For each read *r*, fetch the frequency vectors of all its constitutive k-mers
3. These frequency vectors are matched against the BAC signatures, allowing for a small number of missing/extra pool entries: if no good match exists that frequency vector is discarded

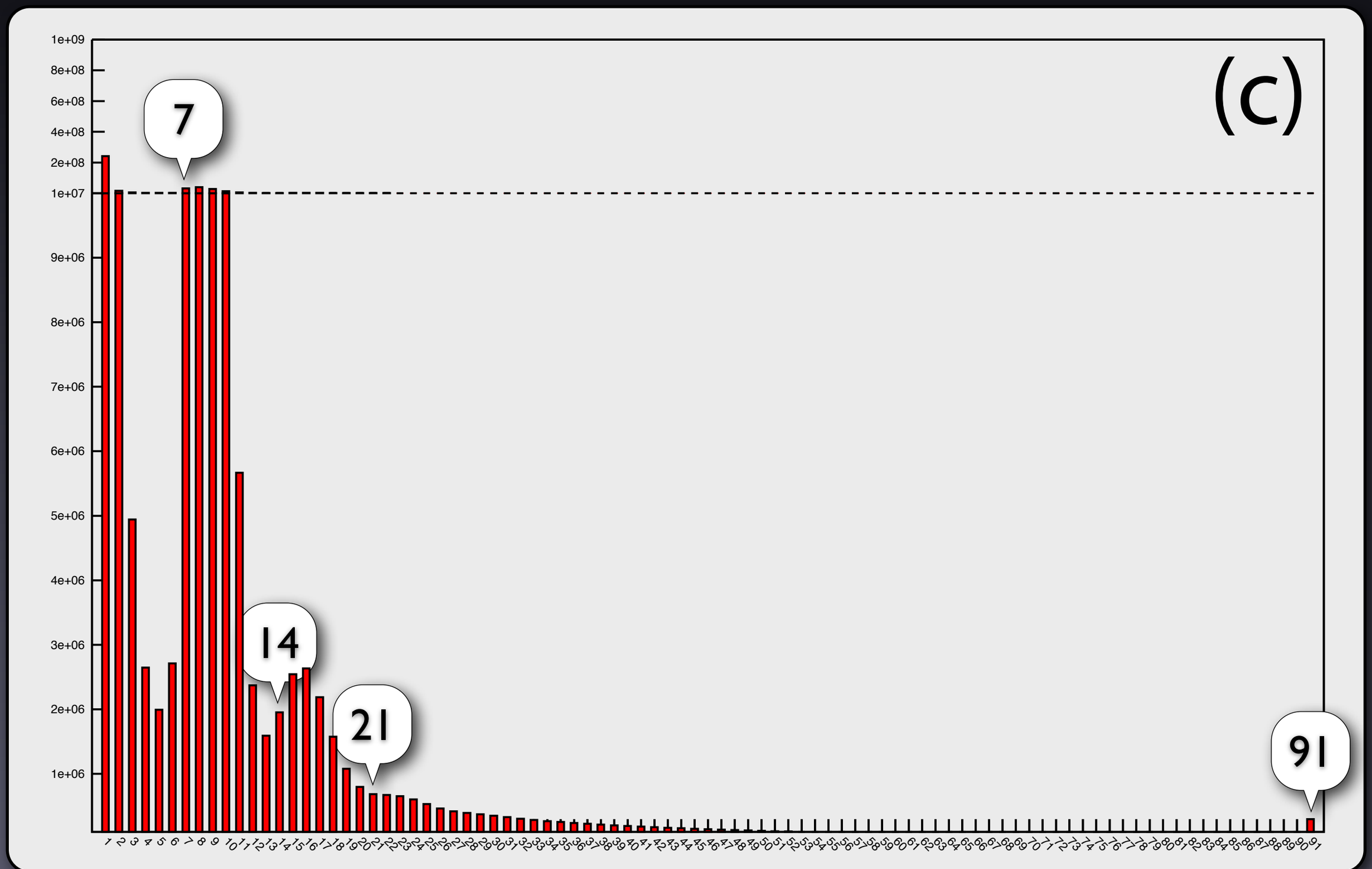
HashFilter's k-mer based strategy

4. Only the frequency vectors that match a valid BAC signature are combined to form the signature of read *r*
5. The read signature is matched again against the BAC signatures to determine the BAC(s) to which *r* should be assigned

Rice k-mer signature size distribution



Barley k-mer signature size



Deconvolution results

- Rice: HashFilter deconvoluted 81.5% of the reads, which translated into an average BAC sequencing depth of ~87x
  [*time*: 164+33+22 min, *memory*: 120 Gb]
- Barley: HashFilter deconvoluted 71.3% of the reads (87% of the usable bases), which translated into an average BAC sequencing depth of ~137x
  [*time*: 340+99+37 min, *memory*: 43 Gb]

Deconvolution accuracy

- Rice: 99.57% of the deconvoluted reads were assigned to either the correct BAC or to a BAC overlapping the correct BAC
- Barley: for 68.7% of the deconvoluted paired-end reads, the left and the right mate were assigned to the same set of BACs despite the fact that HashFilter processed them independently [22% of paired-end reads had one end for which the BAC set was empty]

Assembly

- Velvet assembled individual BACs, for 10 different choices of the hash length parameter [Zerbino *et al.*, *Genome Res.* 2008]
- Recorded the statistics for the assembly that achieved the largest N50 - does not guarantee the 'best' overall assembly
- [N50: the minimum length of all contigs/scaffolds that together account for at least 50% of the target genome]

Assembly statistics

Target	*Size* (Mb)	*Seq. depth*	*% reads used*[c]	*N50* (bp)	*% Sum*
Rice – 1 BAC (perfect deconvolution)[a]	0.151	56x	82.7%	132,865	98.7%
Rice – 1 BAC (HASHFILTER deconvolution)[a]	0.151	87x	82.3%	47,551	90.7%
Rice – 169 BACs (no deconvolution)[b]	26	56x	83.2%	4,236	73.1%
Rice – 2,197 BACs ($k = 25$, no deconvolution)	332	56x	5.9%	1,148	30.6%
Barley – 1 BAC (HASHFILTER deconvolution)[a]	0.129	137x	87.6%	7,210	87.8%
Barley – 169 BACs (no deconvolution)[b]	22	26x	67.1%	4,270	69.5%
Barley – 2,197 BACs ($k = 25$, no deconvolution)	286	180x	25.3%	3,845	56.6%
Barley – whole genome ($k = 31$)	5,300	31x	13.3%	2,857	30.5%

Velvet: rows 1,2,3,5,6; SOAPdenovo: rows 4,7,8
(a) average over 2,197 assemblies
(b) average over 91 assemblies

Quality of BAC assemblies

- Rice: compared the BAC contigs against the "true" sequence; average BAC coverage 76.8%, average gap size 263bp, average # gaps 138, average overlap size 107bp, average # overlaps 75
- Barley: extracted 202 BAC assemblies that were expected to contain certain genes; 90% of them contained the expected gene with an average coverage of ~90%

Final remarks (1/2)

- BAC-by-BAC sequencing/assembly might be necessary for large, highly repetitive genomes
- BAC-by-BAC sequencing on NGS hinges on the ability of multiplexing hundreds of samples; DNA barcoding does not scale
- Combinatorial pooling is cost-effective and practical alternative to exhaustive DNA barcoding (both can be combined)

Final remarks (2/2)

- Experimental results on synthetic rice data and real barley data confirm that the deconvolution process is very accurate
- Resulting BAC assemblies have high quality
- Manuscript submitted, preprint available at http://arxiv.org/abs/1112.4438

Acknowledgements

Botany and Plant Sciences, UC Riverside

Timothy Close (supervision, BACs, libraries, sequencing)

Steve Wanamaker (sys admin, read demux and cleaning)

Prasanna Bhat (Illumina OPA)

Yaqin Ma (sequencing library prep)

Josh Resnik (BAC pooling)

Computer Science, UC Riverside

Stefano Lonardi (supervision, deconvolution, assemblies)

Gianfranco Ciardo (deconvolution)

Denisa Duma (rice synthetic data, deconvolution)

Matthew Alpert (assemblies)

Burair Alsaihati (deconvolution)

Yonghui Wu (Illumina OPA deconvolution)

Serdar Bozdag (compartmentalized physical map)

Computer Science, University of Torino

Francesca Cordero (HashFilter)

Marco Beccuti (HashFilter)