# Pooled Genomic Indexing


## Miklós Csűrös

Department of Computer Science and Operations Research
Université de Montréal

a joint work with

Aleksandar Milosavljevic

Bioinformatics Research Laboratory & Human Genome Sequencing Center

Baylor College of Medicine

## Pooled shotgun reads

developed at Baylor College of Medicine Human Genome Sequencing Center

1. DNA from genomic clones are pooled together

2. shotgun libraries are prepared from the pool

$\Rightarrow$ random reads from a set of clones

usage: sequencing (CAPSS) & physical mapping (PGI)

## Large genome sequencing

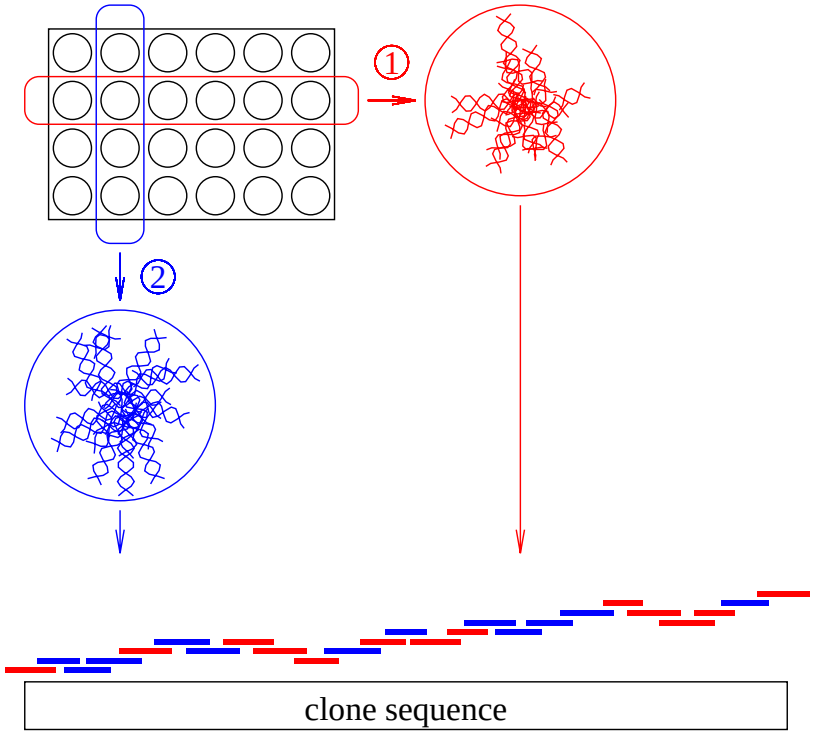
- human (✓)
- mouse and rat (1 yr)
- cow, dog and chicken (?)
- rhesus monkey or chimpanzee or ...
- ...

## Recipes for large genome sequencing

- whole-genome-sequencing — WGS (Celera)
  1. shotgun whole genome
  2. assemble whole genome

- clone-by-clone sequencing — CBC (public)
  1. shotgun clones
  2. assemble clones
  3. assemble genome from clone sequences

# Clone-array pooled shotgun sequencing (CAPSS) — Cai et al. 2001



## CAPSS vs. WGS vs. CBC

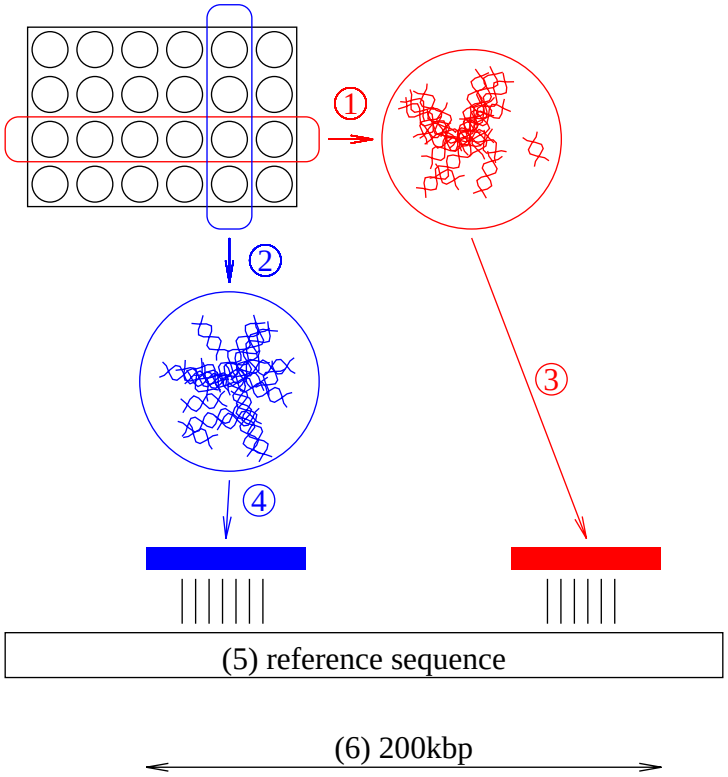
$N$ clones ($N \approx 20$ thousand)

$F$ shotgun reads ($F \approx 50$ million)

| | shotgun libraries | computational subproblem size |
|---|---|---|
| WGS | 1 | $F^2 \approx 2.5 \cdot 10^{15}$ |
| CAPSS | $2\sqrt{N} \approx 300$ | $\left(\frac{F}{\sqrt{N}}\right)^2 \approx 1.25 \cdot 10^{11}$ |
| CBC | $N \approx 20000$ | $\left(\frac{F}{N}\right)^2 \approx 6.25 \cdot 10^6$ |

CAPSS: balance between chemistry and computations

# Pooled Genomic Indexing (PGI): comparative physical mapping of clones



# Indexes

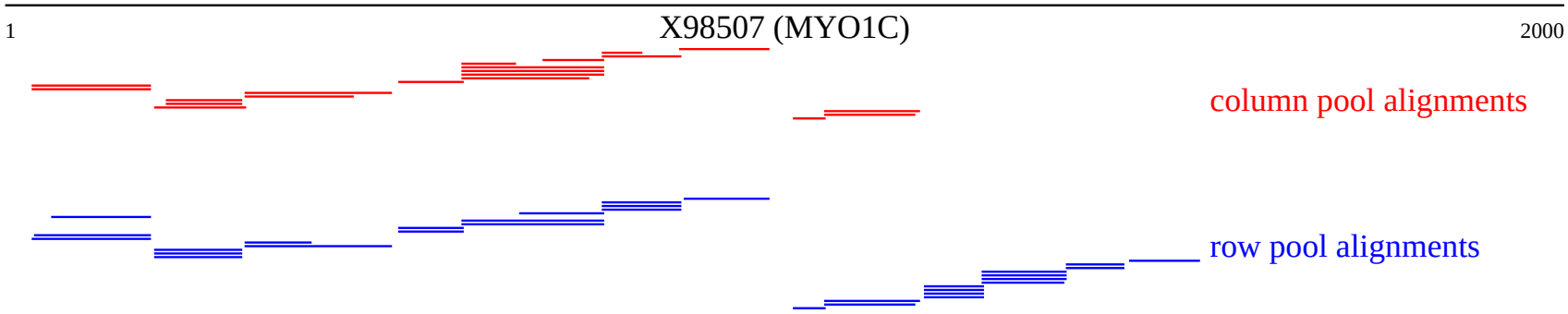
**index:** detected homology between clone and reference sequence

clone-array shotgun reads from row & column pools

index creation: BLAST fragments against reference sequences

close hits from a row & column pool: index to the clone at the intersection

Example



## Mouse experiment

207 mouse phase3 sequences $\geq$ 50kb

$15 \times 14$ array

2X shotgun coverage per clone (121400 simulated reads)

BLAST w/ Unigene, HTDB, Human genome

| | UG | HTDB | HS |
|---|---|---|---|
| # indexes | 723 | 488 | 1472 |
| # indexed clones | 159 (77%) | 139 (67%) | 172 (83%) |

[BLAST thresholds: length $\geq 40$, score $\geq 60$, E $\leq 10^{-5}$]

## Ambiguous indexes

hits from $> 2$ pools on the same reference sequence:
cannot assign a clone to the index unambiguously

| | $C_1$ | $C_2$ |
|---|---|---|
| $R_1$ | $B_{11}$ | $B_{12}$ |
| $R_2$ | $B_{21}$ | $B_{22}$ |

if hits from $R_1$, $R_2$, $C_1$, and $C_2$: $B_{11}$ and $B_{22}$ or $B_{12}$ and $B_{21}$?

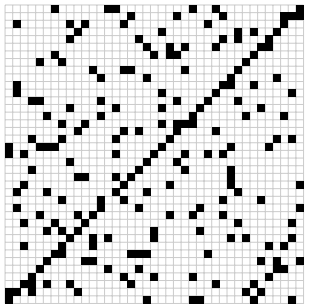
$\longrightarrow$ overlapping clones
$\longrightarrow$ gene families
$\longrightarrow$ repeats

undetected ambiguity: false positive

## Ambiguous indexes: solutions

1. **reshuffled clones** on a new array

2. **sparse array**:
for all choices of two rows and two columns, at most three out of the four cells at the intersections have clones assigned to them



3. **other pooling designs**:
based on Reed-Solomon codes (Kautz & Singleton 1964, Dyachkov et al. 2000)

## Simulation experiments

simple, shuffled, and sparse-array pooling

1. mouse (207 clones): simulated shotgun, simulated pooling
$14 \times 15$ arrays,
$39 \times 39$ sparse array

2. rat (625 clones): real shotgun, simulated pooling
$25 \times 25$ arrays
$87 \times 87$ sparse array

## Mouse experiment

(207 clones, 120 thousand fragments, $2X$ coverage)

Number of correct indexes / false positives

| | UG | HTDB | HS |
|---|---|---|---|
| simple | 723 / 248 | 488 / 108 | 1472 / 69 |
| shuffled | 756 / 76 | 514 / 18 | 1549 / 238 |
| sparse | 823 / 22 | 569 / 11 | 1634 / 69 |

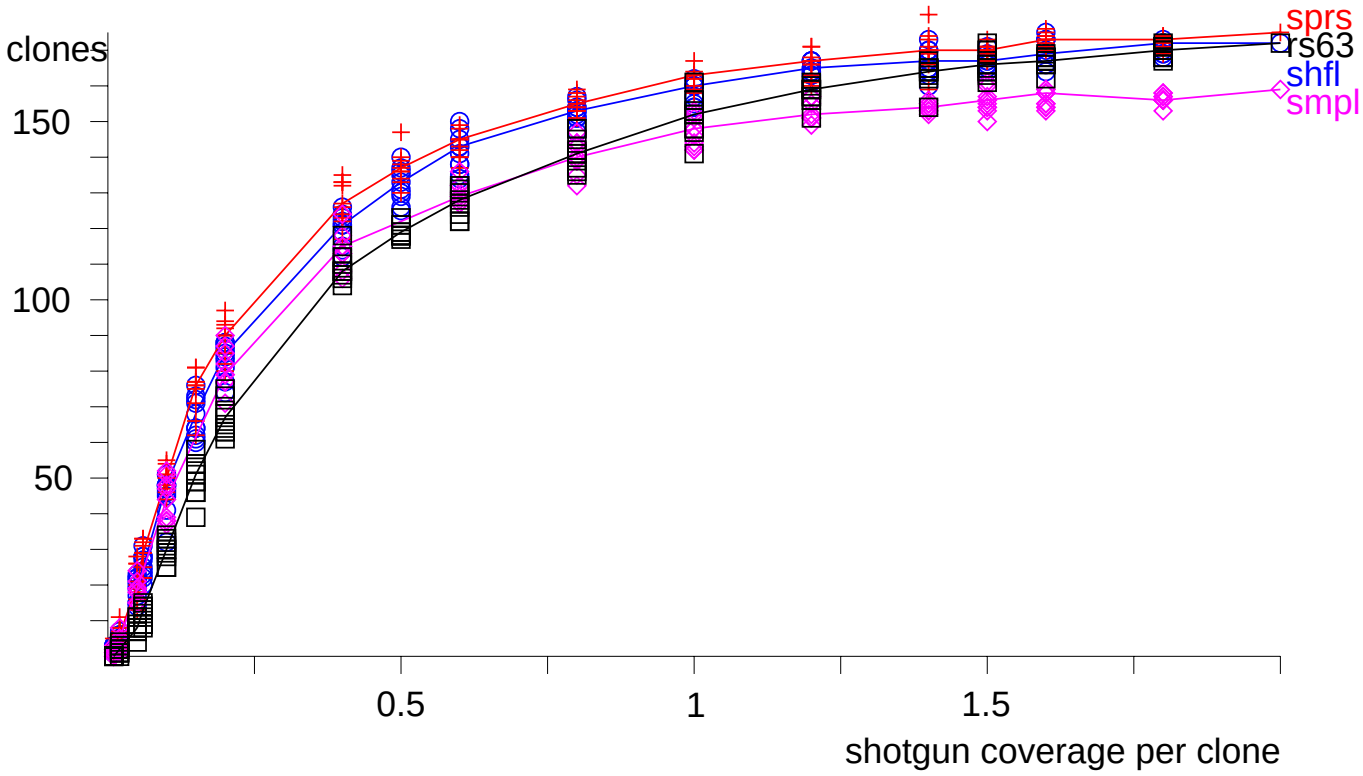
[7–10% of the fragments produce useful alignments]

Number of correctly indexed clones

| | UG | HTDB | HS |
|---|---|---|---|
| simple | 159 (77%) | 139 (67%) | 172 ( 83%) |
| shuffled | 172 (83%) | 150 (72%) | 180 (87%) |
| sparse | 175 (85%) | 152 (74%) | 185 (88%) |

# Mouse experiment — lower coverage

Arrayed pooling against Unigene: number of indexed clones



## Rat experiment

(625 clones, 700 thousand fragments, $1.5X$ coverage)

indexing by Unigene

| | correct indexes | false positives | indexed clones |
|---|---|---|---|
| simple | 1418 | 236 | 384 (61%) |
| shuffled | 1383 | 30 | 409 (65%) |
| sparse | 1574 | 17 | 451 (72%) |

## Theoretical results

1. pooling designs: shuffled and sparse-array pooling

2. probabilistic model for indexing

   Lander-Waterman statistics for coverage $c$

   1. probability of correct indexing
   2. probability of false positives
   3. number of hits within an index

## Sparse-array pooling

use combinatorial geometry; rows correspond to points, columns to lines: clone at the intersection if the point lies on the line.

$\longrightarrow$ for all two rows and two columns, at most three clones.

using finite field $\mathsf{GF}(m)$ w/ $m$ prime power:

rows: $(x, y) : x, y \in \mathsf{GF}(m)$ columns: $(a, b) : a, b \in \mathsf{GF}(m)$
clones: $ax + b = y$

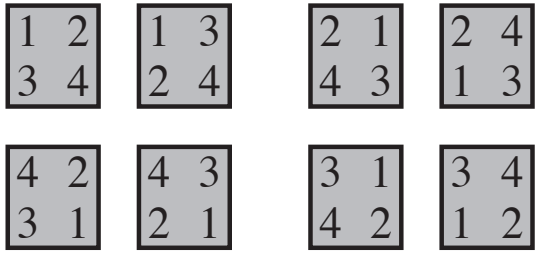
$m^2 \times m^2$ array, $m$ clones per pool, $N = m^3$ clones total

$O(N^{2/3})$ pools for $N$ clones — matches asympototic lower bound

## Shuffling

method: random shuffling

rectangle: two rows and two columns
preserved rectangle: 4 clones in a rectangle after and before shuffling with the same clones on the diagonals




**Theorem.** Expected number of preserved rectangles is approximately $1/2$.

## Random shuffling

**Proof.** Probability that a particular rectangle is preserved:

$$p = \frac{8\binom{m}{2}^2}{(m^2)(m^2-1)(m^2-2)(m^2-3)}.$$

Expected number of preserved rectangles: $\binom{m}{2}^2 p = \frac{1}{2} + \frac{2}{m}(1 + o(1))$.

works also if non-square array (unlike transversal designs)

expected number of preserved rectangles on shuffled $m \times m'$ array

$$\frac{1}{2} + \frac{m + m'}{mm'}(1 + o(1))$$

## Probabilities: homology between a clone and a reference sequence

expected length of random shotgun read: $\ell$

**effective length** $M$: number of positions in which a random fragment produces an alignment

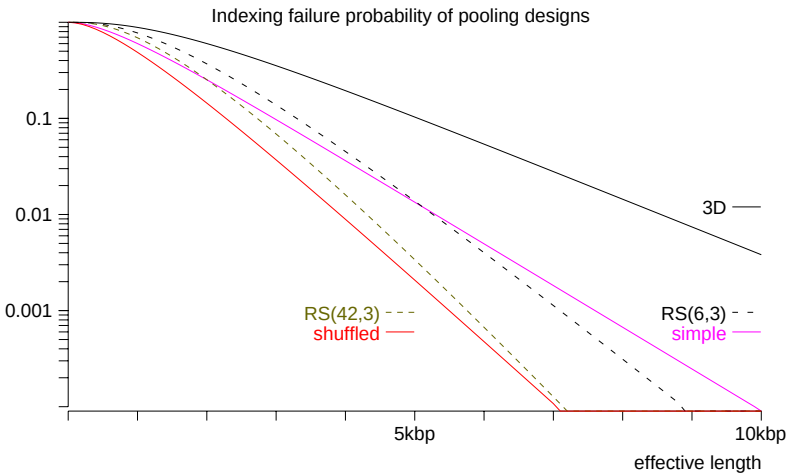
pooling design: clone included in $n$ pools, $k$ out of those identify the clone

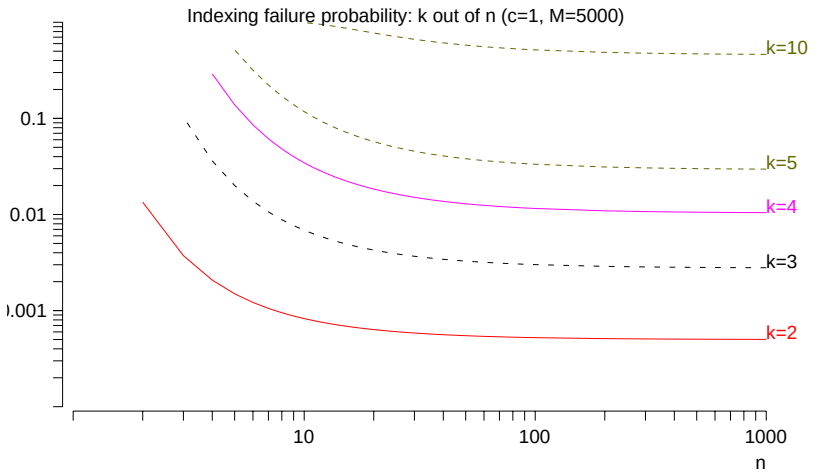
probability of $\geq 1$ alignment within one pool: $p_{\geq 1} \approx 1 - \exp\left(-c\frac{M}{n\ell}\right)$

probability of correct indexing

$$p_M = \sum_{t=k}^{n} \binom{n}{t} p_{\geq 1}^{t} (1 - p_{\geq 1})^{n-t} = 1 - e^{-c\frac{M}{\ell}} \sum_{t=0}^{k-1} \binom{n}{t} \left(e^{c\frac{M}{n\ell}} - 1\right)^{t}$$

# Probabilities for different designs




$$\lim_{n\to\infty}(1-p_M) = e^{-c\frac{M}{\ell}} \sum_{t=0}^{k-1} \frac{\left(c\frac{M}{\ell}\right)^t}{t!}.$$

$\text{Poisson}(c\frac{M}{\ell}) < k$: if $c$ is small, a design with large $k, n$ is not good

## Random concluding remarks

PGI creates indexes from low-coverage shotgun reads: no overhead in sequencing projects, helps directed sequencing

shorter clones (50–100k) are ok for mapping

non-adaptive group testing with control over failure probability (coverage)

physical mapping: ambiguities need not be fully resolved — one identified clone is already good

sophisticated pooling designs are ok but at low coverages, shuffled (transversal design) is best

algorithms for CAPSS sequence assembly and PGI index resolution

compare to hybridization-based mapping: actual sequence information is retrieved (no overhead in sequencing)

compare to BAC-end sequencing: works for inter-mammal indexing, is cheap, and mid-clone sequence info is also obtained

can index genomic or cDNA clones with genomic or transcribed reference sequences within and across species

indexing by homologies between shotgun reads for clone ordering

can reduce necessary coverage 5–10 fold by using very short reads: tags in a SAGE-like approach

comparative map from 0.1–1X coverage: whole genome phylogeny without sequencing

http://www.iro.umontreal.ca/~csuros/

csuros@iro.umontreal.ca