



Tandem Expansions and Other Segmental Rearrangements in Human Genome Evolution

S. Cenk Sahinalp

CENTER FOR
COMPUTATIONAL
GENOMICS

CWRU

now at SFU

Acknowledgements

Sahinalp lab: Can Alkan, Eray Tuzun,
Evan Eichler, Jeff Bailey
Meral Ozsoyoglu, Murat Tasan

NSF (BioI, TofC & IDM)

Charles B. Wang Foundation

Ohio Board of Regents (PRI)

Crash Course: Human Genome Evolution

DNA sequence: a contiguous substring of the genome

Measuring evolutionary/functional relationship of sequences S, R:

Similarity score: insert “-” symbols to S, R to maximize:

$$D(S',R') = \sum -\log d(S'_j,R'_j)$$

$d(S'_j,R'_j)$: probability of mutation between *aligned* characters S'_j,R'_j
per given year

$d(x,y) \sim 1.5 \cdot 10^{-9}$ for non-functional DNA

Percentage similarity score:

$$P(S,R) = 100 \cdot h(S'_{OPT},R'_{OPT})/|S'_{OPT}|$$

Repeat (of a sequence S): sequence R whose percentage similarity score with S is “high”.

Duplication: The evolutionary process of *copying* a substring S elsewhere.
Repeats are generated by duplication events. 60% of Human Genome is composed of repeats.

Significance of Genome Repeats

- Key to proper assembly of the human genome (& that of other species currently being sequenced).
- *~60% of the genome sequence is repeated* - tandem or interspersed (>1Kb segments <30% divergence)
- Duplicated/missing regions contain genes. Excess/lack of gene segments result in *genomic diseases: birth defects* (frequency >0.1%) & *adult diseases: cardiovascular disease, osteoporosis, etc.*
- Duplications + point mutations: key to genome evolution
- Repeats are key to mechanisms for segmental rearrangement: *replication slippage, retrotransposition...*

How do repeats influence genome assembly

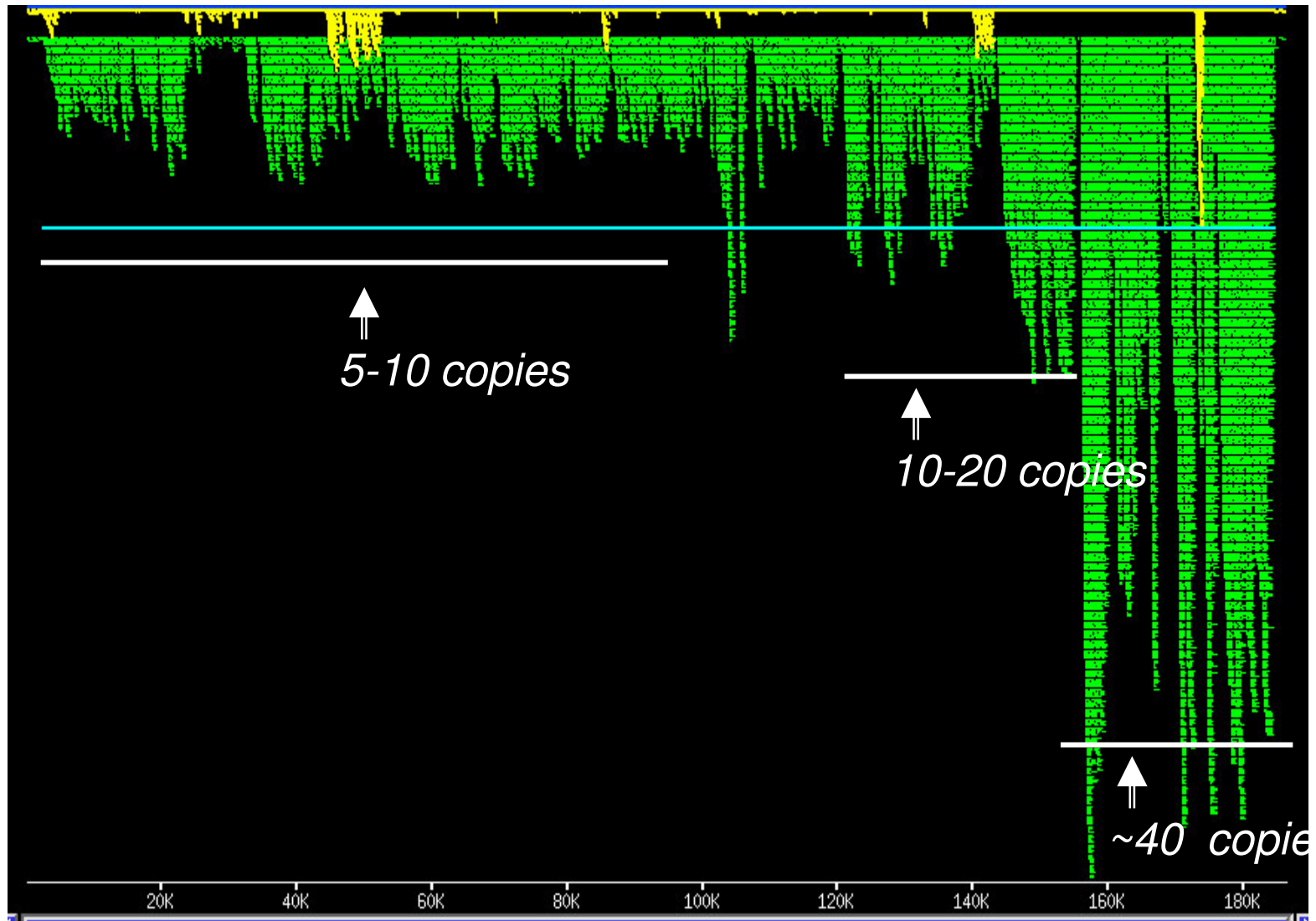
What you want



is not necessarily...What you get



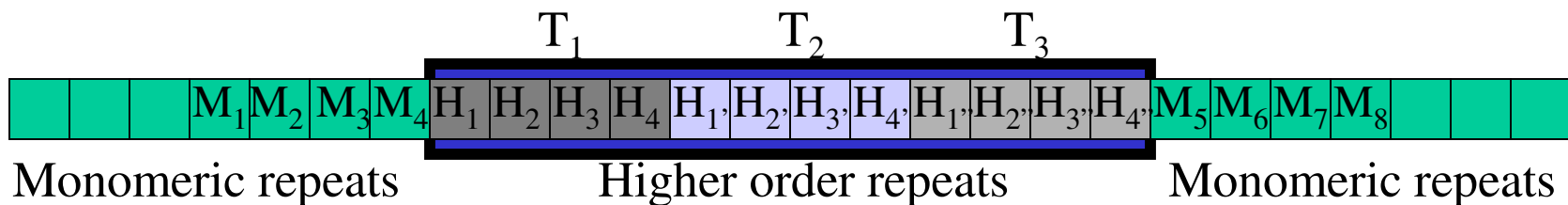
HGP (AC002038) vs Celera: duplications >98%



Tandem Repeats

- Very common in Human genome; especially satellite DNA
- **Primary example:** alpha-satellite sequences common to all human chromosomes
- **Basic repeat unit:** ~171bp monomer, O(1000) occurrences in each chromosome
- Much of alpha-satellite DNA can be grouped in “higher order” repeat units of k-monomers (k=4-20, fixed for each chromosome)
- **Avg divergence btwn monomer pairs:** 20-40%
- **Avg divergence btwn high-order repeat units:** 5%

Alpha-satellite DNA organization



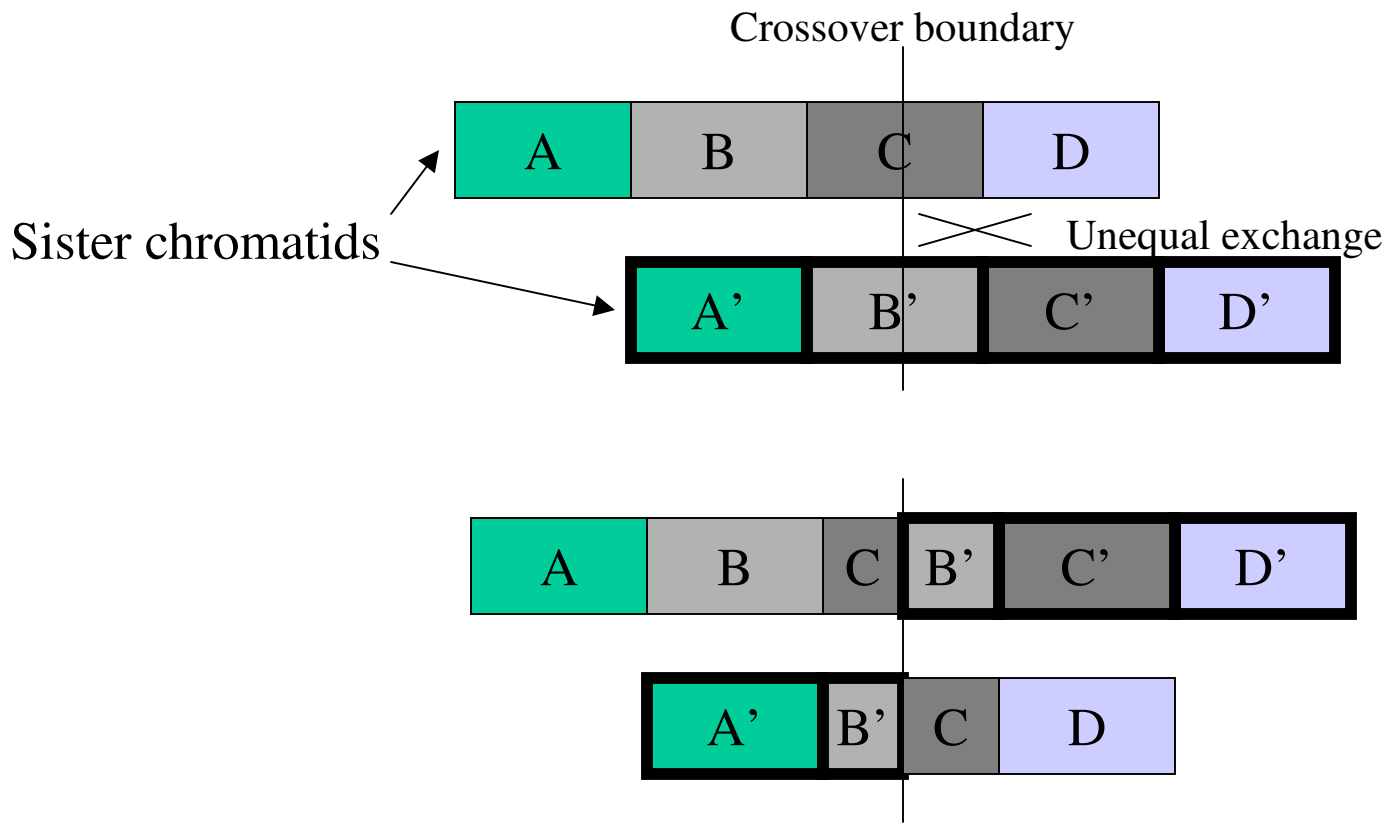
$\text{Divergence}(H_i, H_j) = \text{Divergence}(T_i, T_j) = \sim 5\%$

$\text{Divergence}(H_i, H_j) = \text{Divergence}(M_i, M_j) = \text{Divergence}(M_i, H_j) = \sim 20\%$

Earlier tandem amplifications in monomeric units

Later amplifications in higher order k-mers

Mechanisms for Tandem Amplification: Unequal crossover



Unequal crossover for alpha satellite DNA amplification

[Smith'76]:

Unequal exchange between sister chromatids during meiosis may provide the key mechanism for satellite DNA amplification

Amplification first occurs in single monomeric units

But once some $k > 1$ units are amplified, the next amplification will tend to involve k units again!

Thus the higher order.

What's new?

Due to repetitive nature, alpha-satellite DNA is very difficult to shotgun sequence + assemble.

[Willard&Waye'87, Mashkova-et-al'98, Alexandrov-et-al'01]: sequenced significant portions of alpha-satellite DNA – identified consensus for each monomer position in higher order units

[Alexandrov-et-al'01]: a complementing mechanism to unequal crossover may have transposed the higher order sequences from one source to other chromosomes – overtaking the function of monomeric structure

Our goals vs our means

Verify if unequal crossover provides the sole mechanism for alpha-satellite DNA evolution by using the following data:

Built a library of monomers from sequenced higher order repeat regions from available literature

Used the monomer library + repeatmasker to identify BAC-clones from HGP databases involving alpha satellite DNA

Extracted all monomers from each clone involved.

Note: Location of a monomer within the clone

OR

Location of a clone within the chromosome
can not be reliably known (satellite DNA is replicative)

Thus we have 33 clones (mostly draft sequences), each involving a set of monomers whose locations within a clone is unknown

Available algorithmic methods do not apply

[Benson&Dong'99, Berard&Rivals'02, Elemento-et-al'02, Jaitly-et-al'02, Zhang&Wang'02]:

heuristics + approximation algorithms + hardness of computing the “most likely”/”least costly” sequence of amplifications – under the **assumption that unequal crossover is the sole mechanism for the expansion** of tandem array

[Tang-et-al'02]: given the phylogenetic tree of an ordered list of monomers

Does the positional ordering of monomers agree with their phylogenetic ordering?

i.e. is it possible that the tandem array could be generated by unequal crossovers only?

Our methodology

Identified each clone as:

higher order, monomeric, mixed

Constructed the phylogenetic trees of:

all monomers from each clone

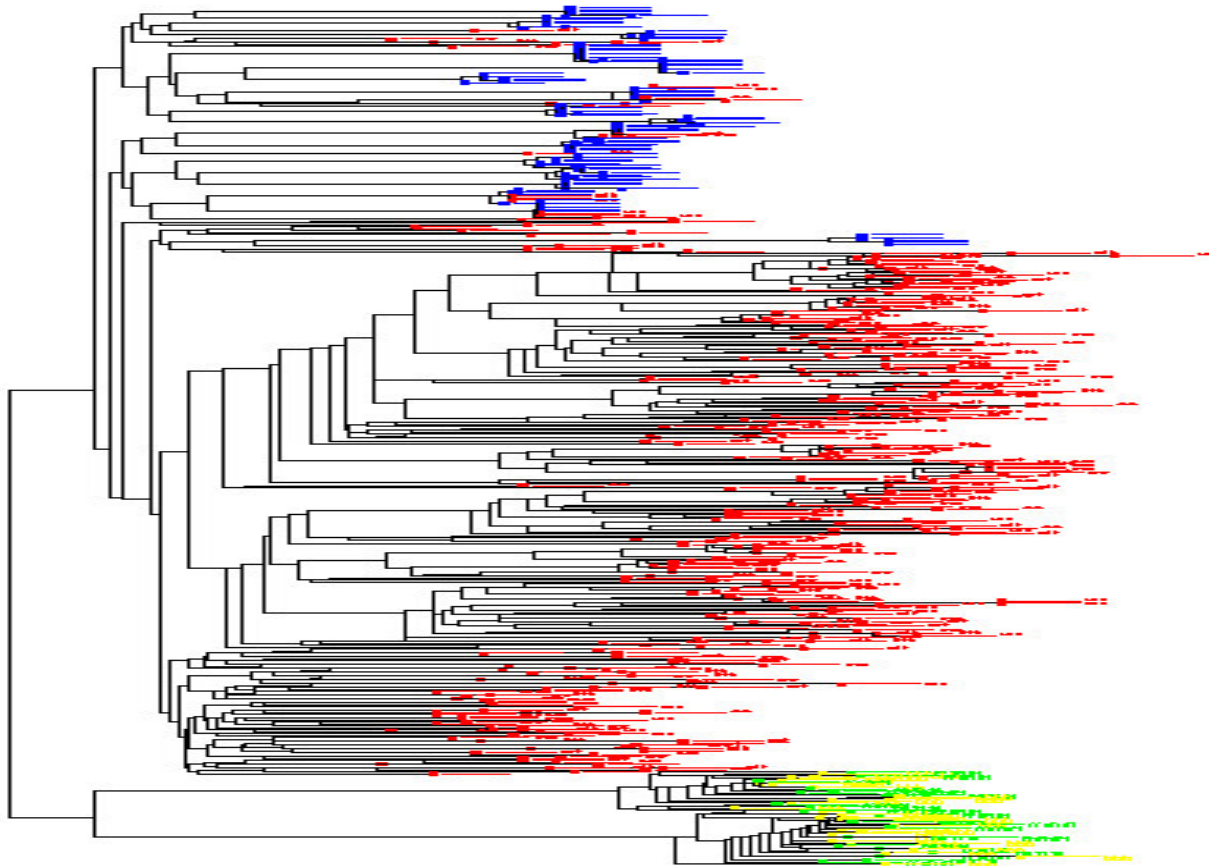
-against-

all monomers from the higher order library

Observations:

- (1) **strongly separated**: higher order repeats from the library vs monomeric repeats from clones
- (2) **mix well**: higher order repeats from the library and higher order repeats from the clones
- (3) **mix well**: monomeric repeats from different clones

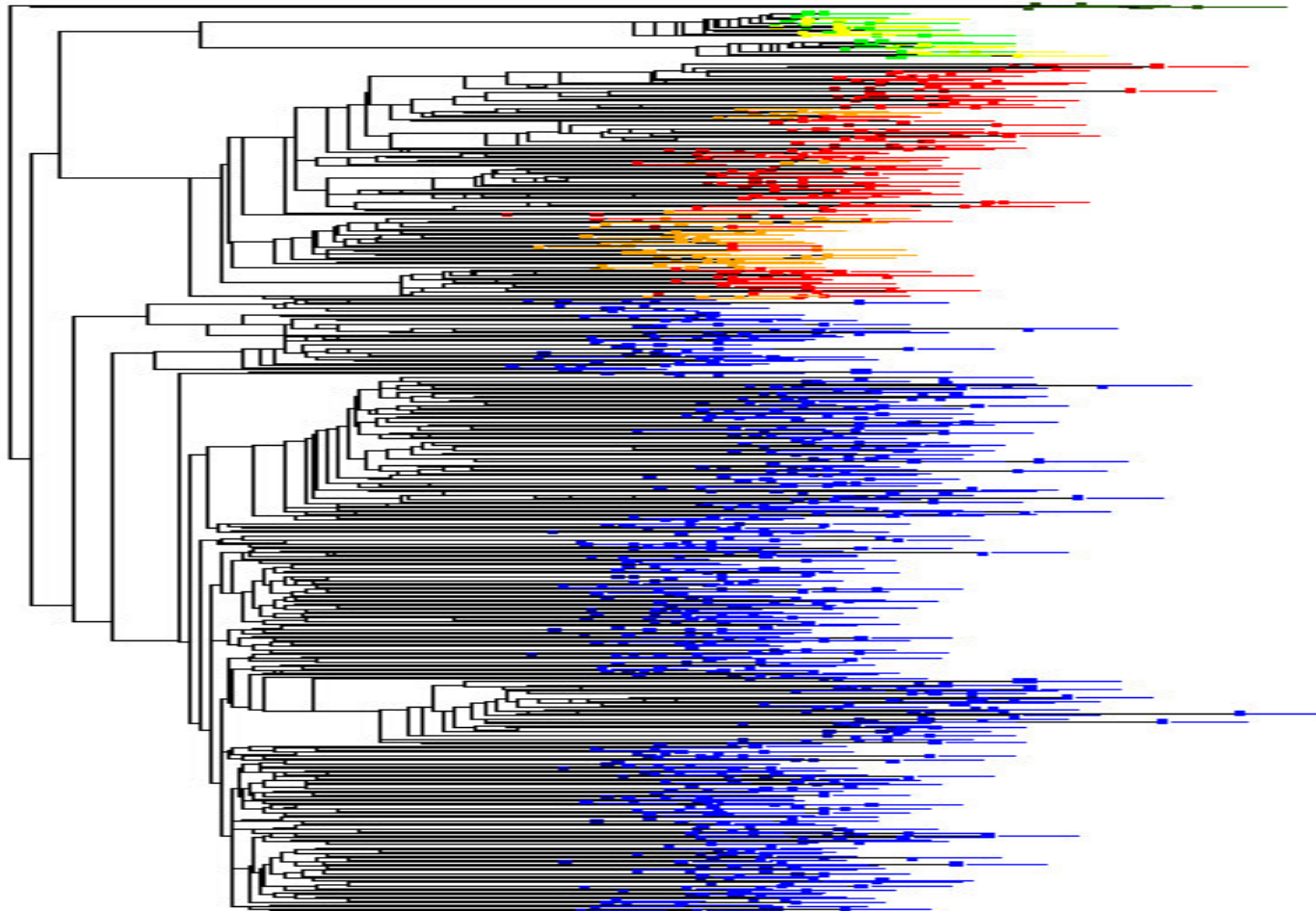
All monomers from higher order clone
vs
All monomers from higher order repeat library



All monomers from monomeric clone AC026005

vs

All monomers from higherorder repeat library



Algorithmic question

What is the likeliness of evolutionary separation between the monomers of a (monomeric) clone and those from the higher order region of the same chromosome?

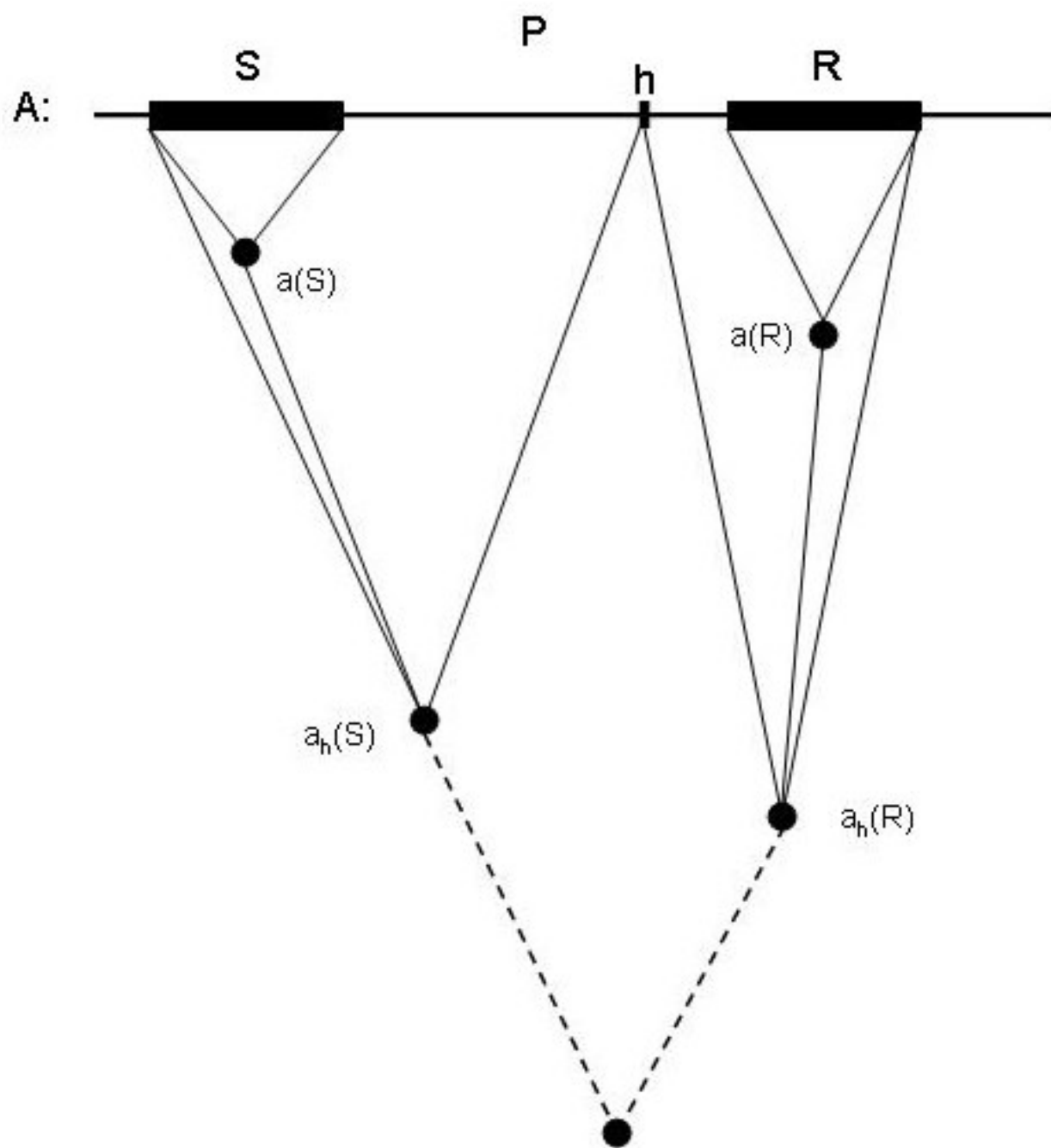
i.e. what is the probability that given a tandem repeat array, two independently picked subarrays (one higher order, one monomeric) have evolve separately?

Assumptions:

Amplification by exactly one monomer at a time in the evolution of the monomeric subarray [Tang-et-al'02]

At each step of amplification, the crossover boundary is distributed uniformly

The direction of amplification is also uniform



Probability of evolutionary distinctness

$\Pr(E \mid v, w, k)$: probability of two subarrays of lengths (in terms of monomers) v and w , with distance k can have unique lowest common ancestors

Lemma:
$$\Pr(E \mid v, w, k) = \frac{k + 1}{k + v + w - 1}$$

Lemma:

If k is distributed uniformly at random to $[0 \dots (m - w - v)]$ then:

$$\Pr(E \mid v, w) \cong 1 - \left(\frac{v + w - 2}{m - v - w + 1} \cdot \ln \frac{m - 1}{v + w - 1} \right)$$

Our observations

All higher order clones mix well with the higher order repeat library – the library seems to be comprehensive

5 out of 11 monomeric clones were evolutionarily distinct from high order repeat library;

i.e. $\text{freq}(E) = 5/11 = .45$

For $w=150k/171$, $v=1$, $m= 200k/171$

$$\text{Pr}(E|v,w) = .14 < 3.\text{freq}(E)$$

Probability of observing 5 events of evolutionary distinctness out of 11 independent experiments:

$$\sum_{i \geq 5} \binom{11}{i} (.14)^i (1-.14)^{11-i} \leq 0.01$$

PART 2: Distance Based Indexing for Sequence Similarity Search

Overview

Problem:

Develop efficient tools for Sequence Similarity Search.

Sequence similarity measures:

Character edit distance, **block edit distance**, weighted variants - capturing evolutionary and functional relationships between genome/protein sequences.

State of the Art (Beyond BLAST):

Exact sequence similarity search suffers from the Curse of Dimensionality: Lower bounds: exponential time in preprocessing or querying in the worst case, even for Hamming distance [Borodin-et-al'99]

Approximate Sequence Similarity Search poly-time only for Block edit distance [Muthukrishnan-Sahinalp'00] with approx-factor $O(\log n)$.
No other result with subexponential running time available.

Our approach

Question 1: Worst case is bad! (as anybody using BLAST knows)

BUT - can we do better for well behaved data sets?

Are data sets of practical interest well behaved?

Question 2: In sequence spaces dimensions do not have a clear meaning
– most traditional indexing techniques are not applicable.

Even if data is “well behaved” the only available approach is distance based indexing (VP, MVP trees, etc.).

Can we use distance based indexing for sequence proximity search?

Our contributions

Answer 1: Protein sequences, genome sequences, etc. have very regular (polynomial or exponential) pairwise distance distributions.

Answer 2: Regularity in pairwise distance distributions can be exploited by distance based indexing methods.

Outline

Sequence Similarity Measures – an overview:

character & block edit measures, weighted versions

Distance Based Index Structures:

VP trees and modifications to almost metrics

Exploiting properties of data sets for improving VP trees

Preliminary results:

human proteome, world languages, synthetic sequences

Applications for Sequence Similarity Search under Edit Distances

Computational sequence analysis in genomics, proteomics:
similarity between DNA, RNA and protein sequences
indicate functional and evolutionary relationship

Data compression:

by textual substitution

Time series analysis & data mining

Information retrieval

The distance measure used should capture the notion of
similarity related to the application domain.

Character Edit Distances

Edit operations allowed: insertion, deletion, replacement of characters – *to capture simple mutations.*

(Levenshtein) edit distance: minimum number of (unweighted) edit operations to transform one string to another,
standard DP solution in $O(n^2)$ time.

In general: each edit operation has a fixed cost (independent of context)
If two strings have a common origin, the cost of an edit operation may indicate $-\log(\text{probability})$ of that edit operation in a fixed time interval
[PAM, Blossum]

The most likely sequence of edit operations are in fact the least costly sequence of edit operations.

Block Edit distances

- **Edit operations allowed:** block copies, block deletes, block moves, block reversals + all character edits:
to capture segmental rearrangements + mutations
- **Transformation distance** [Varre et.al'99] : minimum number of block edits to transform one sequence to another.
[Muthukrishnan-Sahinalp'00]: NP-hard, $O(\log n)$ factor approximation in $O(n)$ time.
- **Compression distance** [Li et.al'01,'03]: compressibility of one string when the other is available as dictionary.
 $O(n)$ time computable by [Rodeh et.al.'81]

Similarity search problem

Given a set of sequences $X = \{x_1, \dots, x_n\}$,

a distance function $d(.,.)$,

a search radius r , and

a query point q ,

retrieve all sequences that are within distance r to the query sequence.

$$\{x_i \mid x_i \in X \text{ and } d(x_i, q) \leq r\}$$

Distance Based Indexing

Inherently different from Spatial Indexing, or
Multidimensional indexing:

Here *only* the *relative* distances are used for index
construction via space partitioning, and search,
i.e. *no absolute* spatial information on the data elements are
considered.

VP-Tree [*Burkhard-Keller'73, Uhlmann' 91, Yianilos'93*],
GNAT [*Brin'95*], **MVP-Tree** [*Bozkaya-Ozsoyoglu'97*],
M-Tree [*Ciaccia et.al'97*].

Distance based indexing has been defined only for Metric Distances

A metric space X is defined by a distance function

$d: X^2 \rightarrow R$ s.t. for all x, y, z in X ,

- $d(x, y) = d(y, x)$.
- $d(x, y) \geq 0$ and $d(x, y) = 0$ iff $x = y$.
- $d(x, y) + d(y, z) \geq d(x, z)$

$d(.,.)$ is a metric distance function.

VP-Tree

Binary tree that recursively partitions data space using distances of data points to randomly picked vantage points.

Internal nodes: $(x_{vp}, M, R_{ptr}, L_{ptr})$

M : median distance of among $d(x_{vp}, x_i)$ for all x_i in the space partitioned.

x_{vp} : Vantage point.

Leaves: References to data points.

Proximity Search in VP Trees

Given a query point q , a metric distance $d(.,.)$ and a proximity radius r ,

Find all data points x such that $d(x,q) \leq r$.

If $d(q, x_{vp}) - r \leq M$ recursively search inner partition.

If $d(q, x_{vp}) + r \geq M$ recursively search the outer partition.

Else search both partitions.

Levenshtein edit distance and Transformation (block edit) distance are both metric distances.

Weighted edit distances where weights indicate – $\log(\text{probability})$ of edits (mutations) are metrics.

BUT: arbitrary weighted character edit distances and Compression distances are *not* metrics (triangular inequality not satisfied)

However: Both distances are **almost metrics**, i.e., reflexive, symmetric, and satisfy the triangular identity within a constant factor k .

i.e. for all s, r, q in X , $d(s, r) \leq k \cdot [d(s, q) + d(q, r)]$

For compression distance $k=3$.

Distance Based Indexing for *Almost Metrics*

q : *query element*

r : *query radius*

x_{vp} : *vantage point*

M : *median distance value for M*

$d(x,y)$: *almost metric distance function (satisfies triangular inequality within factor k).*

Then,

If $d(x_{vp},q)+r < M/k$ then search the inner partition only.

If $d(x_{vp},q)-k.r > k.M$ then search the outer partition only.

Exploiting properties of data sets

$f(r)$: number of string pairs with distance at most r

We considered

1. Declaration of Human rights in 52 Eurasian Languages – *under compression distance*
2. The complete set of protein sequences active in Brain cells (from SwissProt) – *under character edit distance*
3. Complete human proteome – *under character edit distance*
4. Synthetic sequences with random edit operations

Properties of Textual Data under Compression Distance

Data set:

Declaration of Human rights in 52 Eurasian languages
[Benedetto et.al'02, Li, et.al'01,'03]

We observed exponential distribution:

$$f(r) = k \cdot c^r$$

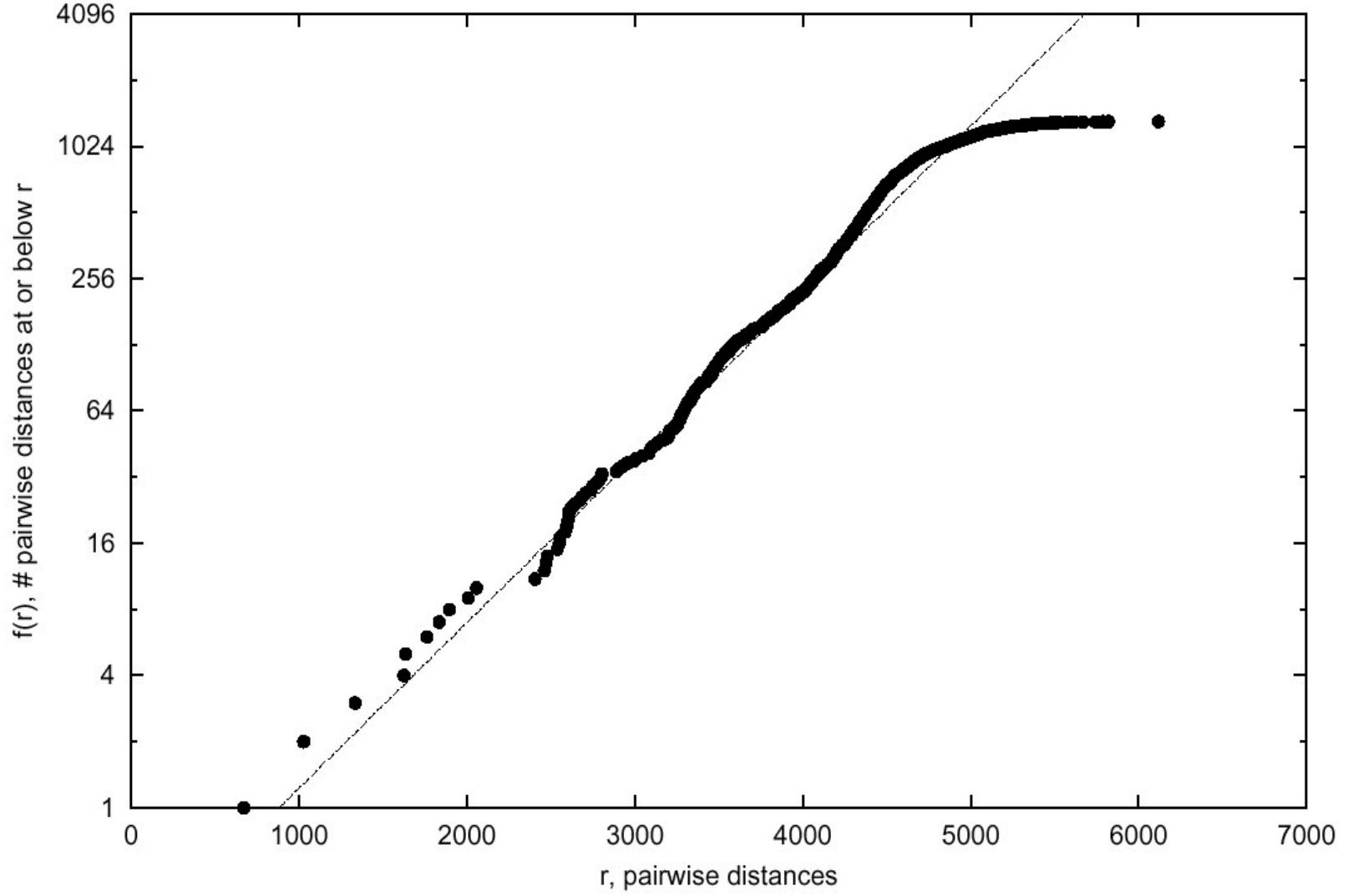
$$\log f(r) = \log k + r \cdot \log c$$

k and c are constants:

$$c = 2^{1/400}$$

$$k = 2^{-2.2}$$

pairwise language distances under limited transformation distance
overlaying an exponential function $f(r)=kc^r$: $c=2^{(1/400)}$, $k=1/(2^{2.2})$



Properties of Proteome data

Data set:

Complete set of protein strings that are active in the brain cells of Humans and other organisms - from SwissProt (93 proteins)

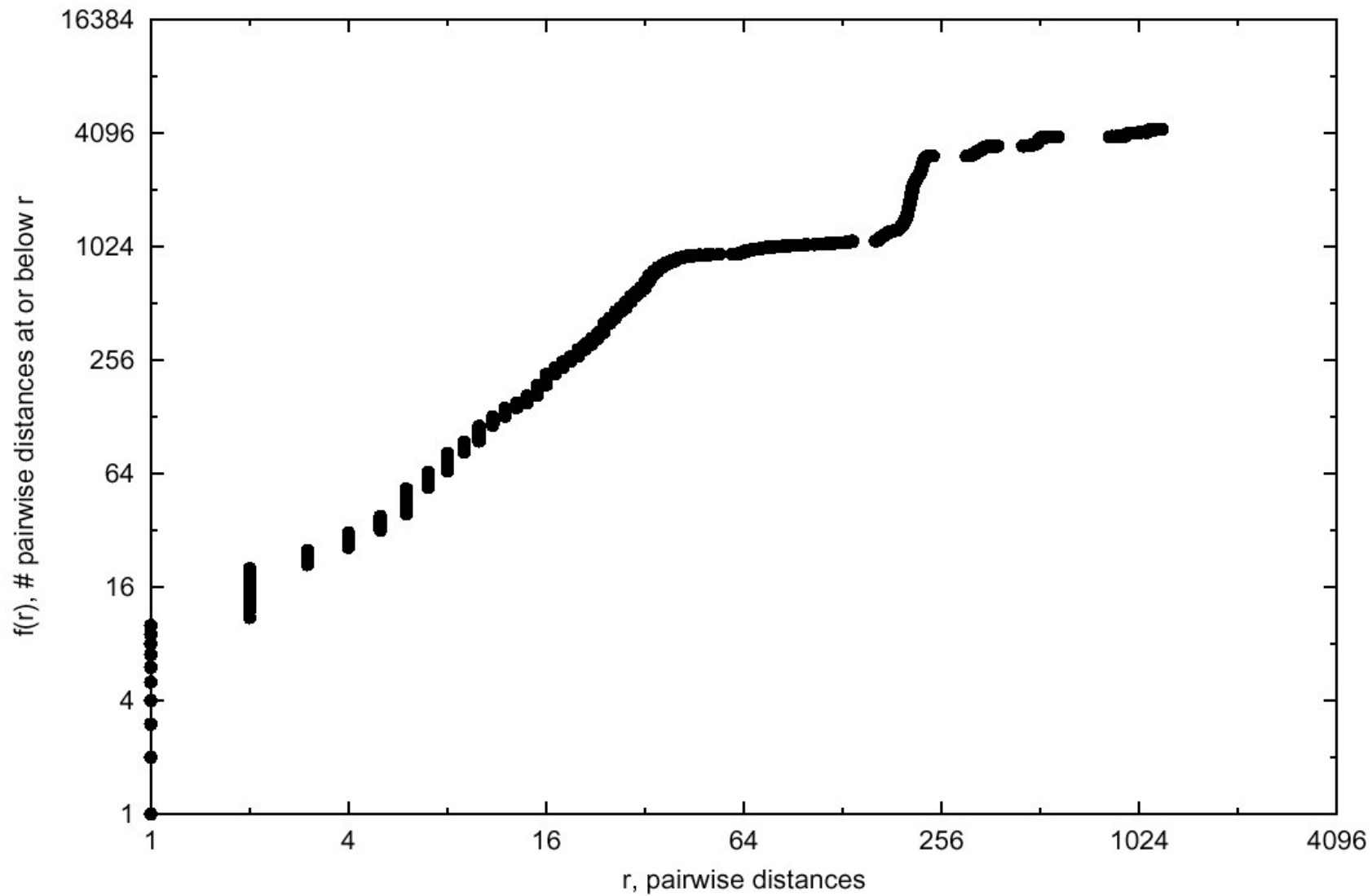
We observed polynomial distribution (power law) under
Levenshtein edit distance (as well as compression distance):

$$f(r) = k \cdot r^c$$

$$\log f(r) = \log k + c \cdot \log r.$$

k and c very similar for the two distance measures.

pairwise protein distances under unweighted character edit distance





pairwise protein distances under limited transformation distance



VP trees for nearest neighbor search

If $f(r) = k \cdot r^c$ (exponential distribution) chances of pruning “inner partition” is negligible – and is ignored

Optimal partition for m points still at the median M of m points w.r.t. the vantage point.

- If $\log c \cdot d(x_{vp}, q) < \log m/2$ then search only the inner partition will occur with probability $k/2$ ($= 2^{-3.2}$ for textual data).
- Otherwise iteratively re-partition the data set according to a new vantage point.

Probability of failing to eliminate the outer partition for $2 \cdot j/k$ vantage points is at most $p = 1/e^j$

Space: $O((2/k)^{\log m})$

Query time: $O(2/k \cdot m^{\log 1+p})$

Experimental Evaluation of VP trees

Pruning efficiency (i.e. number of pairwise string comparisons to respond to a query) in modified VP-trees.

Synthetic data: 2000 strings obtained by performing random (but non-uniform) edits on an initial “query” string:
high degree polynomial distribution under transformation distance.

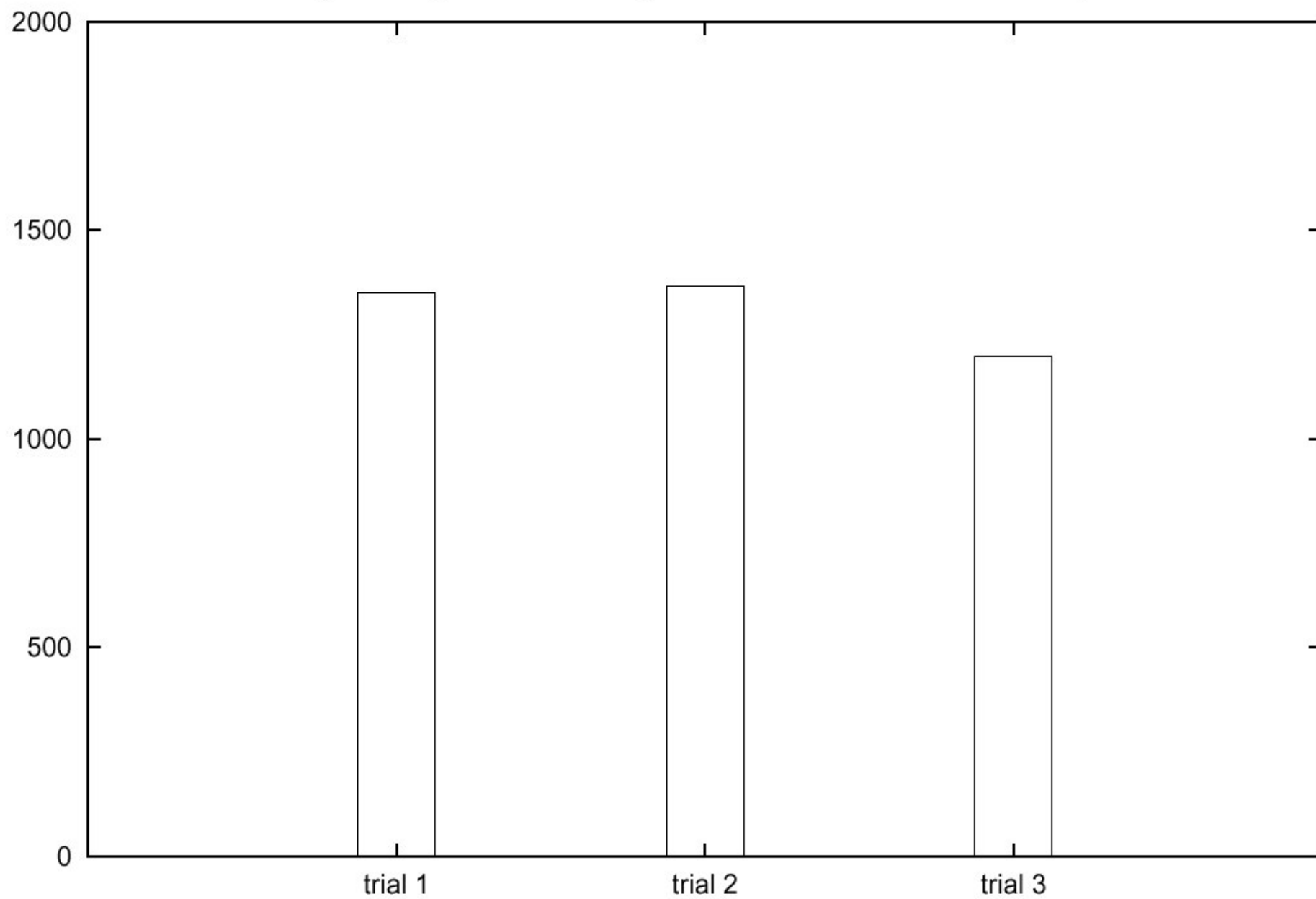
Exact search results for nearest sequence: **90% pruning**

When $k=3$, search results for nearest sequence: **33-45% pruning**

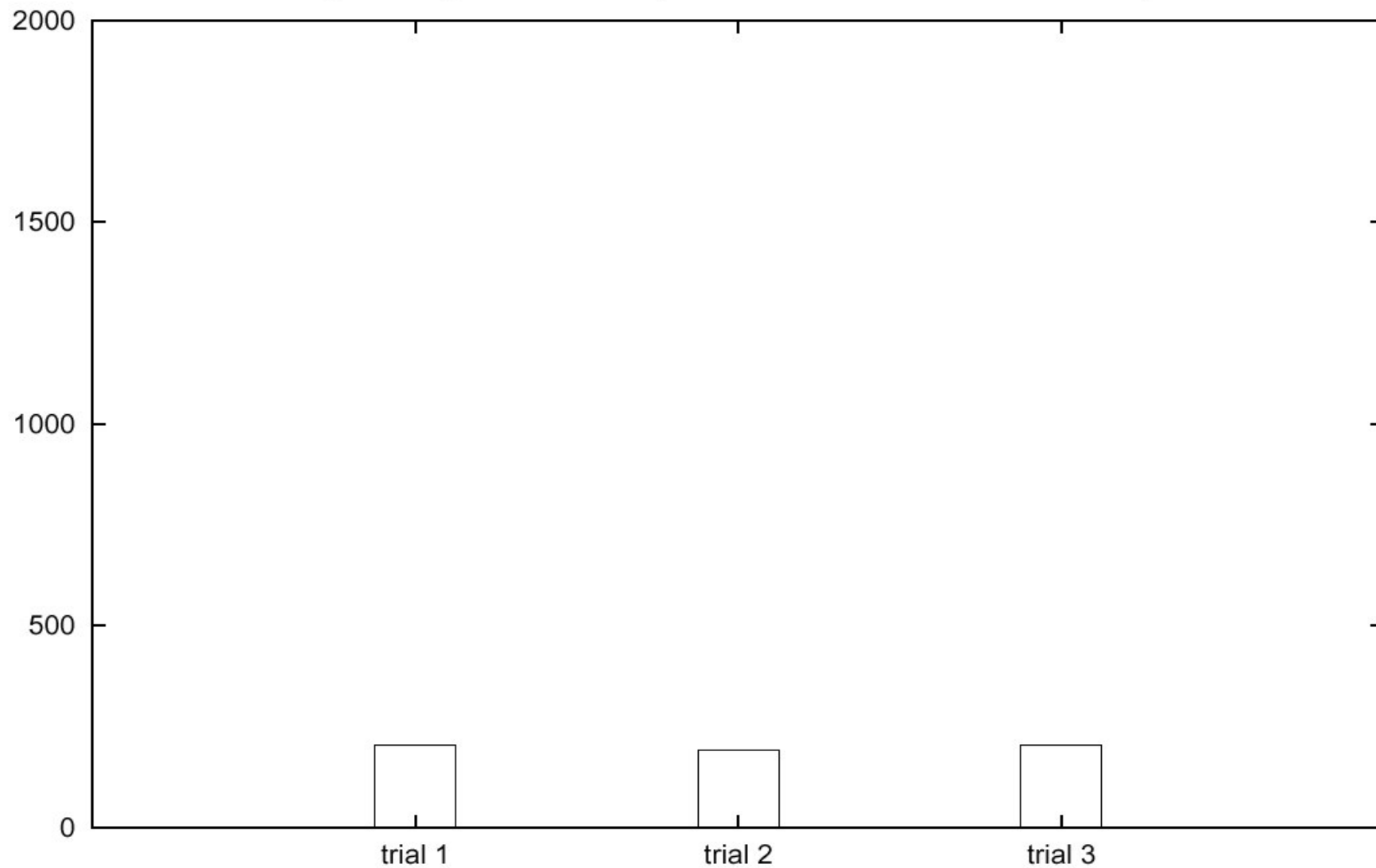
Protein Data: set of active and potential proteins derived from the complete human genome sequence database from Celera (32K):
expected to be exponentially distributed under Levenshtein edit distance.

Search results vary w.r.t. the protein searched.

Number of distance computations for top 5 search of 2000 sequences in search space strings with approximation compression distance in 'almost' metric space



Number of distance computations for top 5 search of 2000 sequences in search space strings with approximation compression distance in assumed metric space



character edit computations for a random sample
set of protein sequences with varying query range

