Aalto University
School of Science

Size Matters: Finding the Most Informative Set of Window Lengths

Jefrey Lijffijt [1], Panagiotis Papapetrou [1 2], and Kai Puolamäki [1]

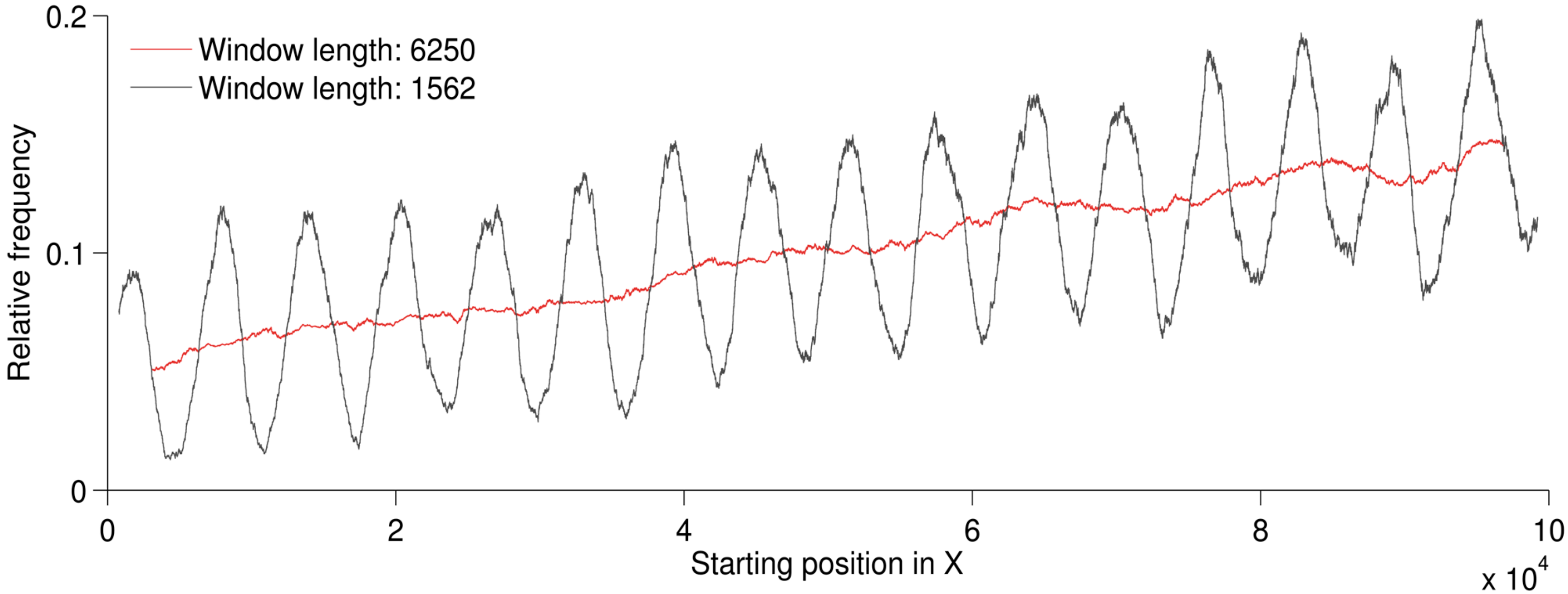
[1] Aalto University, Finland

[2] Birkbeck, University of London, UK

Summary

- Many sequence analysis algorithms use sliding windows
- **Problem: how to choose the length of the window**
- Novel problem setting and approach
- Solution: use several window lengths that can ‘predict’ all
- Solution can be computed efficiently
- Method works well

Example



- Sequences often contain variability at different levels
  - E.g., multiple time scales, daily and weekly rhythm
- Statistic = relative event frequency
- Trends: slow increase and periodic increase/decrease

Related Work

- Solutions are wide-spread (for citations see the paper)

1. User has to choose
2. Optimize towards some objective
   – Fixed-length
   – Variable-length
   – Backing-off
   – Time-fading model (weighting)
   – Many others
3. Use all possible lengths

- None consider optimizing a set of window lengths

Input Data

- *Event sequence X*
  - Fully ordered sequence of events
  - E.g., with four symbols: *ABBCDDAAADCACCABB*

$$X = x_1, \ldots, x_n,\ x_t \in \sigma$$

- *Subsequence $X_j(i)$*
  - Sequence of length *j* starting at index *i*

$$X_j(i) = x_i, \ldots, x_{i+j-1}$$

- *Statistic $f(X_j(i))$*
  - Measure of interest
  - E.g., relative event frequency, or the type/token ratio
  - Can be any algorithm

$$f(X_j(i)) = \#q\ occurs\ in\ X_j(i) / j$$

$$f(X_j(i)) = \#types\ in\ X_j(i) / j$$

Retain Information / Predict All

- Given a set of window sizes Ω

$$\Omega = [\omega_1, \ldots, \omega_m]$$

- Goal: provide as much information wrt $f(X_\omega(i))$ for all ω
  - Using k window lengths

- That is, we want to predict $f(X_\omega(i))$ for all window lengths
  - Based on values $f(X_{\omega 1}(i))$, …, $f(X_{\omega k}(i))$

Window-Trace Matrix

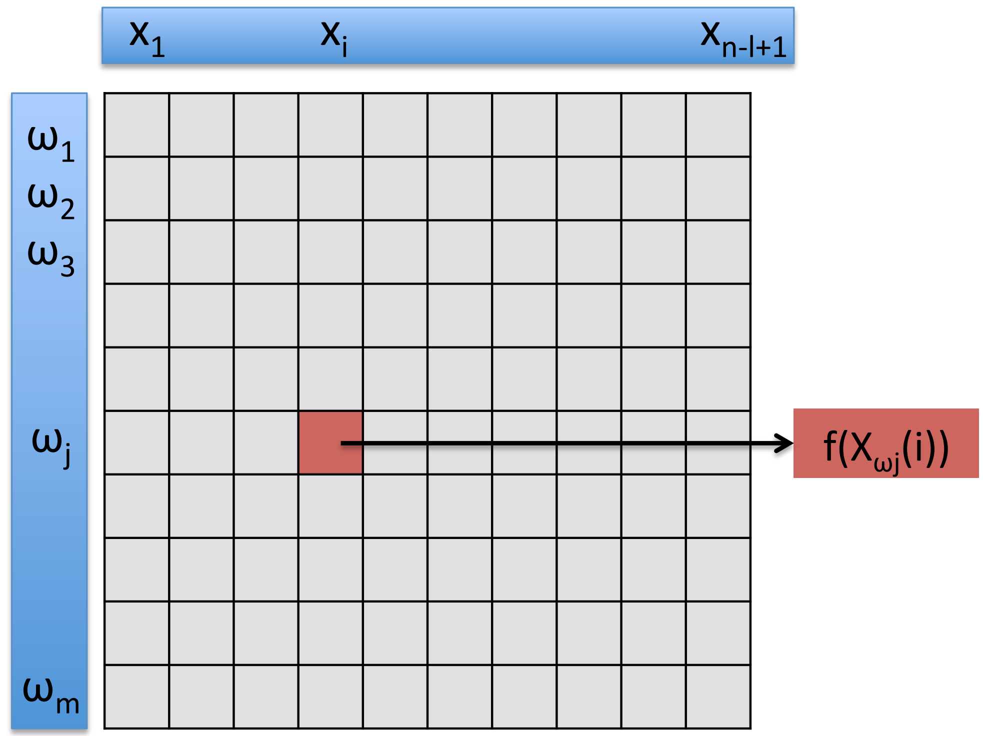
- Given a set of window sizes *Ω*

$$\Omega = [\omega_1, \ldots, \omega_m]$$

- The Window-Trace matrix *T* contains all *f(X*$_{\omega j}$*(i))*

$$T_{ji} = f(X_{\omega_j}(i))$$

- We compute only *N* of the columns



Problem Statement

- *Problem 1 (Maximal variance).* Given a discrete sequence X, find a set $R = \{\omega_1, \ldots, \omega_k\}$ of k window lengths that explain most of the variation in X, i.e., find a set R that minimizes

$$\sum_{\omega_i \in \Omega} \min_{\omega_j \in R} d(\omega_i, \omega_j)$$

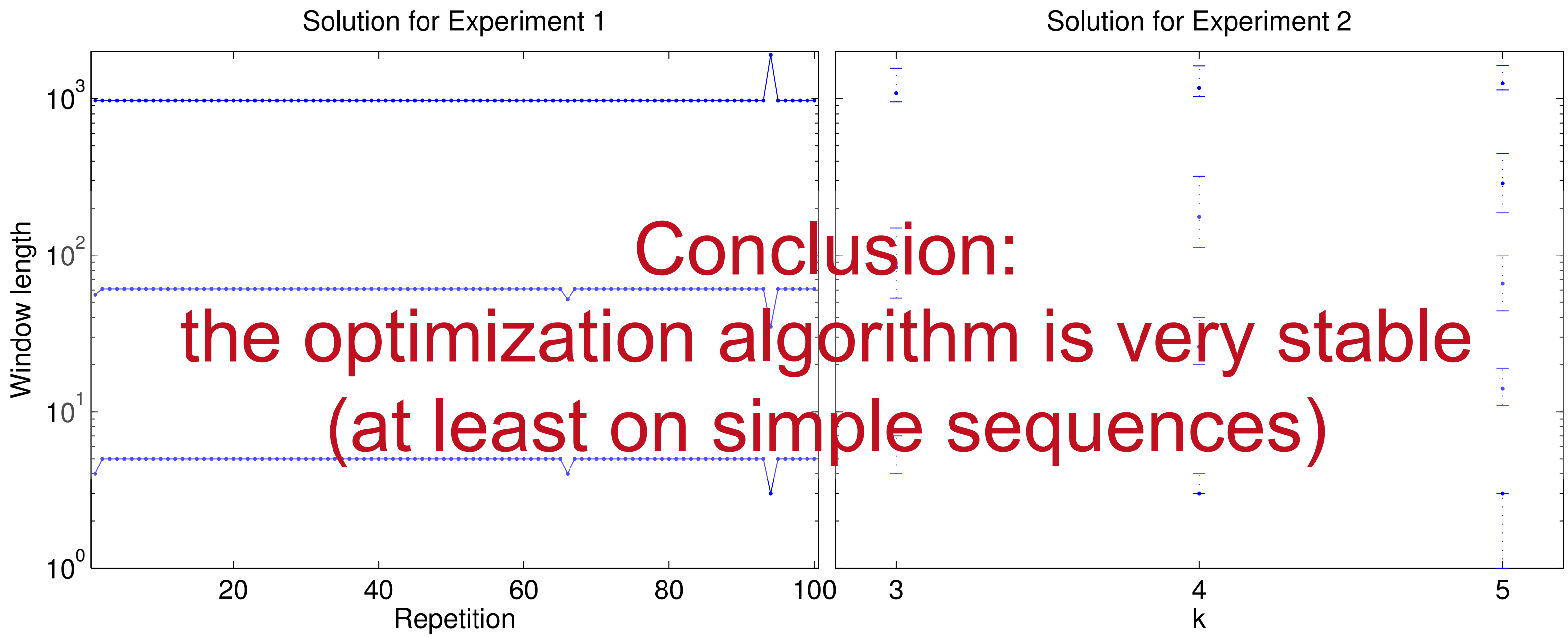
- The distance function that we use is squared error

$$d(\omega_i, \omega_j) = \sum_{k=1}^{n-l+1} (T_{ik} - T_{jk})^2$$

Method

- Problem 1 is equivalent to the k-medoids problem
- NP-Hard (Aloise et al. 2009)
- Optimization algorithm:
  - Compute k-means clustering using Lloyd's algorithm
  - Include in *R* the window lengths closest to each centroid
  - Repeat *r* times and choose best solution (smallest error)
- Computational complexity: $O(r \cdot i \cdot k \cdot N \cdot m)$

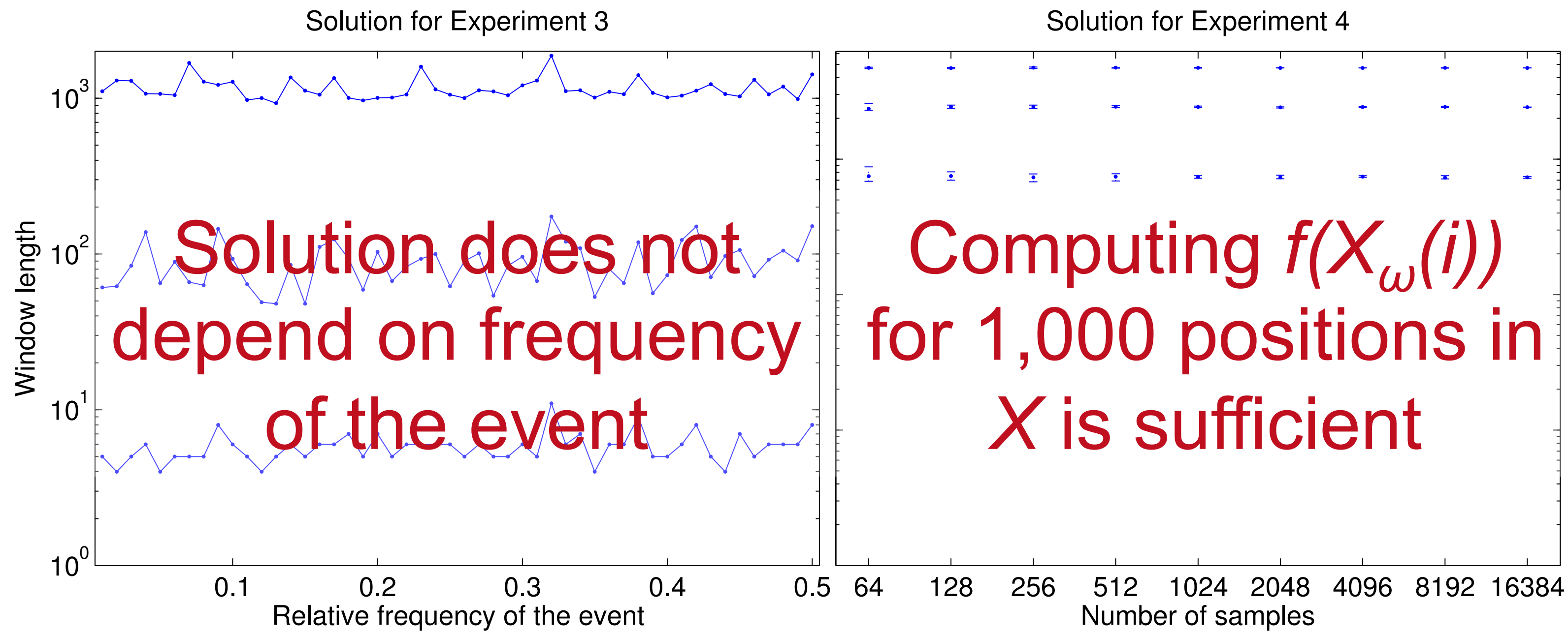
Experiments on Synthetic Data (1/3)



Conclusion:
the optimization algorithm is very stable
(at least on simple sequences)

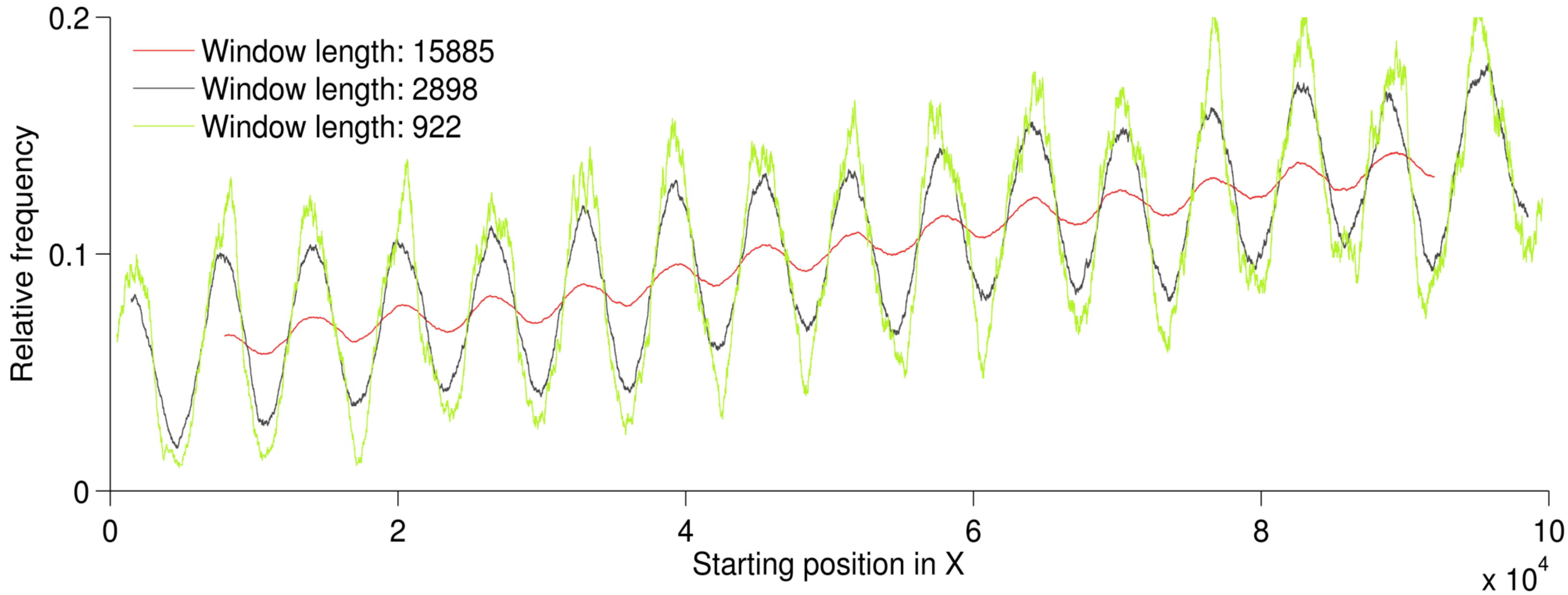
- We studied the solution stability on Bernoulli sequences
  - Length = 10,000, p = 0.1
  1. Repeated runs on one sequence
  2. Repeatedly generate sequences

Experiments on Synthetic Data (2/3)



- We studied the solution stability on Bernoulli sequences
  - Length = 10,000, p = 0.1
  3. Dependency on event frequency
  4. Dependency on number of samples (columns)

Experiments on Synthetic Data (3/3)

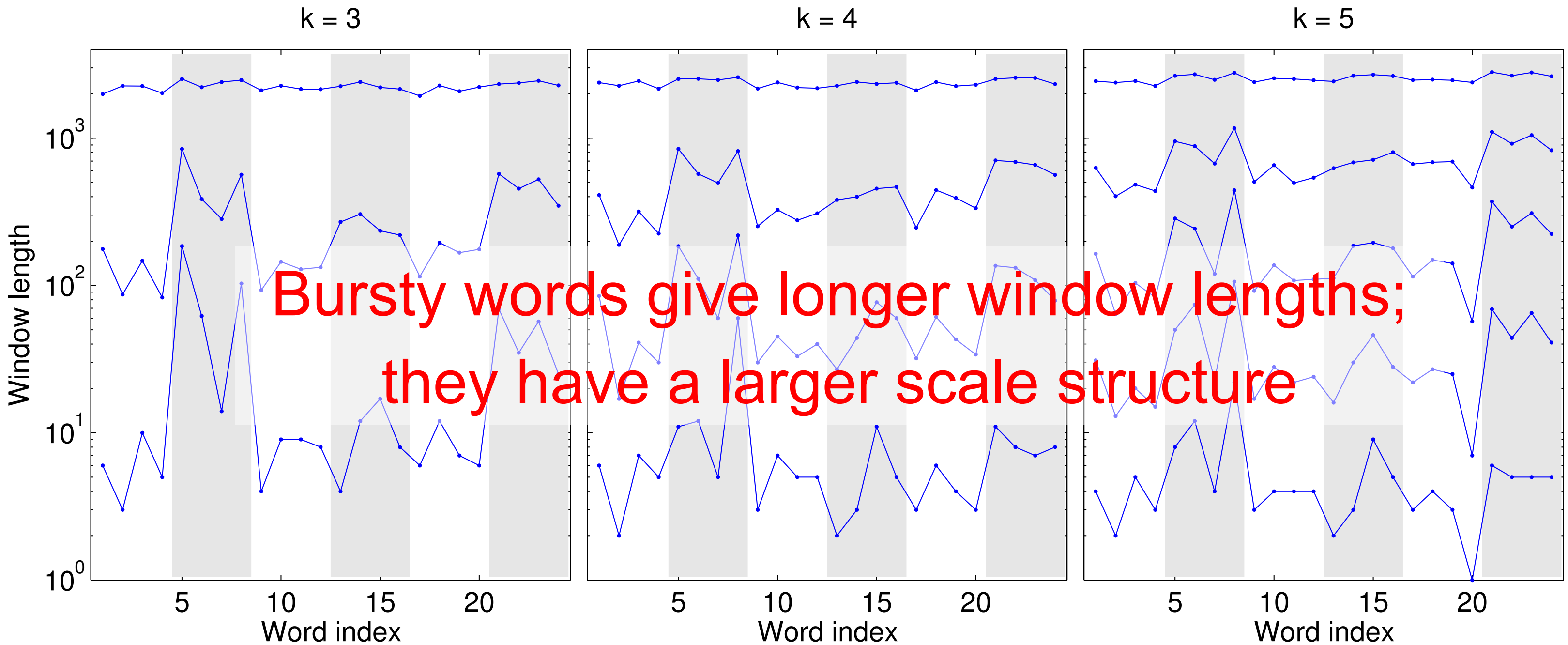


- k = 3 solution for sequence shown at introduction
- Both trends clearly visible
- We can accurately estimate all other window lengths

Burstiness of words in Pride & Prejudice

- A word is *bursty* when it occurs in bursts and lulls
  - Areas with elevated and with lowered frequency
- Non-bursty words: the, and, a, in
- Bursty words: I, you, how, our
- We model burstiness using inter-arrival times
  - IAT: space between two consecutive occurrences of a word
  - Burstiness defined by MLE of Weibull β

Burstiness of words in Pride & Prejudice



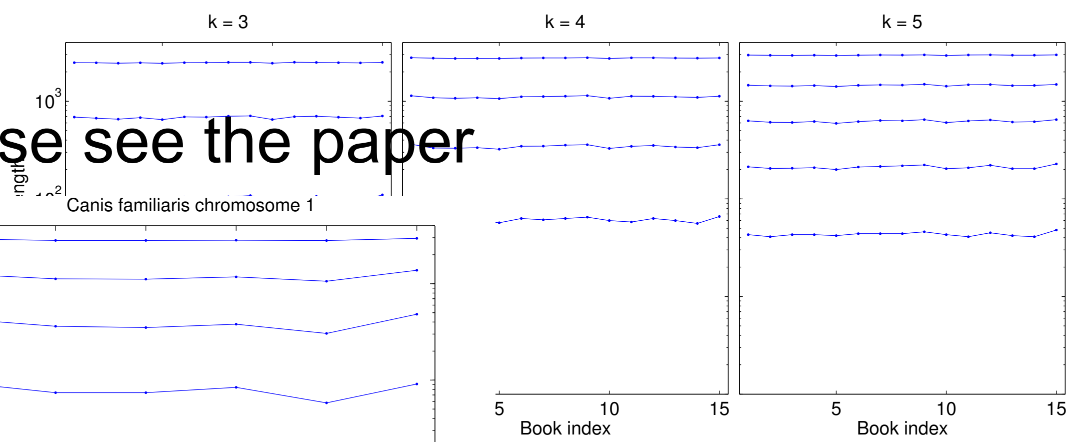
Freq.	Non-bursty	Bursty
Low	met, rest, right, help (1-4)	write, de, William, read (5-8)
Med	time, soon, other, only (9-12)	lady, has, can, may (13-16)
High	with, not, that, but (17-20)	you, is, my, his (21-24)

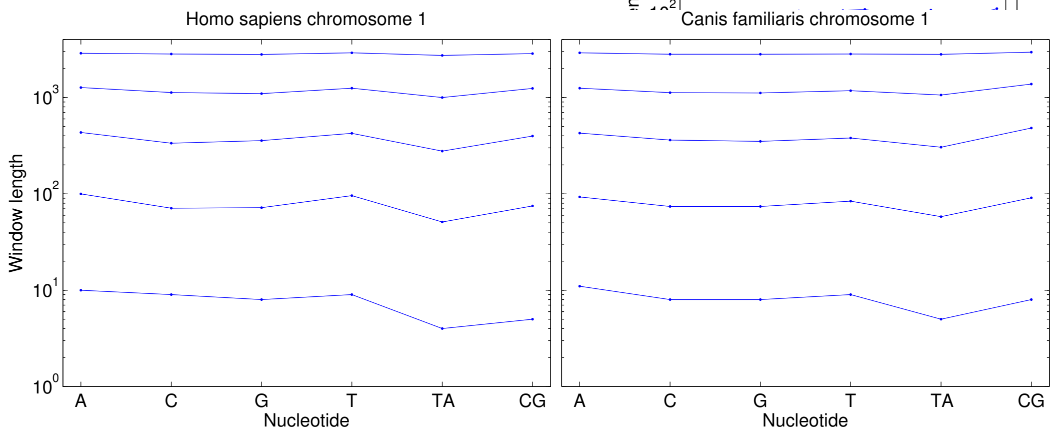
Other Experiments

- Type/token ratio throughout several novels

- Frequency of (di-)nucleotides in DNA

- For the results please see the paper





Summary

- Many sequence analysis algorithms use sliding windows
- **Problem: how to choose the length of the window**
- Novel problem setting and approach
- Solution: use several window lengths that can ‘predict’ all
- Solution can be computed efficiently
- Method works well