The Complexity and Application of Syntactic Pattern Recognition Using Finite Inductive Strings

Elijah Myers
Paul S. Fisher
Keith Irwin
Jinsuk Baek
Joao Setubal

Finite Inductive Strings

- A finite inductive string is one that:
  - Consists of symbols selected from a finite alphabet of possible symbols
  - Eventually terminates and is therefore finite in length
  - Can be characterized using inductive reasoning (i.e. observations of the specific occurrences of symbols can be applied generally to define the entire string)

Factoring

- Given a sample string: aacacgacgt
  - Append the start symbol: Saacacgacgt
  - Create a storage structure (ruling) that uniquely identifies each symbol

Ruling	
S → a	cac → g
Sa → a	cacg → a
aa → c	ga → c
aac → a	gac → g
ca → c	gacg → t

Factoring

- Given the residual string: agagt
  - Append the start symbol: Sagagt
  - Create a second ruling that uniquely identifies each symbol

Level 0 Ruling	
S → a	ca → c
Sa → a	ga → c
aa → c	

Level 1 Ruling	
S → a	ag → a
a → g	gag → t

Factoring

- Given the residual string: t
  - Append the start symbol: St
  - Create a third ruling that uniquely identifies each symbol

Level 0 Ruling	
S → a	ca → c
Sa → a	ga → c
aa → c	

Level 1 Ruling	
S → a	ag → a
a → g	

Level 2 Ruling	
S → t	

Following

- Given a sample string: aacacgaccat
  - Append the start symbol: Saacacgaccat
  - Apply the first ruling to identify those elements that do not conform to the ruling

Level 0 Ruling	
S → a	ca → c
Sa → a	ga → c
aa → c	

Following

- Given the residual string: agacat
  - Append the start symbol: Sagacat
  - Apply the second ruling to identify those elements that do not conform to the ruling

Level 1 Ruling	
S → a	ag → a
a → g	

Following

- Given the residual string: cat
  - Append the start symbol: Scat
  - Apply the third ruling to identify those elements that do not conform to the ruling

Level 2 Ruling	
S → t	

  - With all rulings applied, the final residual "cat" is the portion of the string that did not match the pattern of the factored string

Performance

- Definition of Terms
  - *b* – number of symbols in the alphabet (i.e. base)
  - IB – inductive base
  - N – length of the examined string (number of characters)
  - L – number of levels factored or followed
  - R – number of rules in a given ruling level

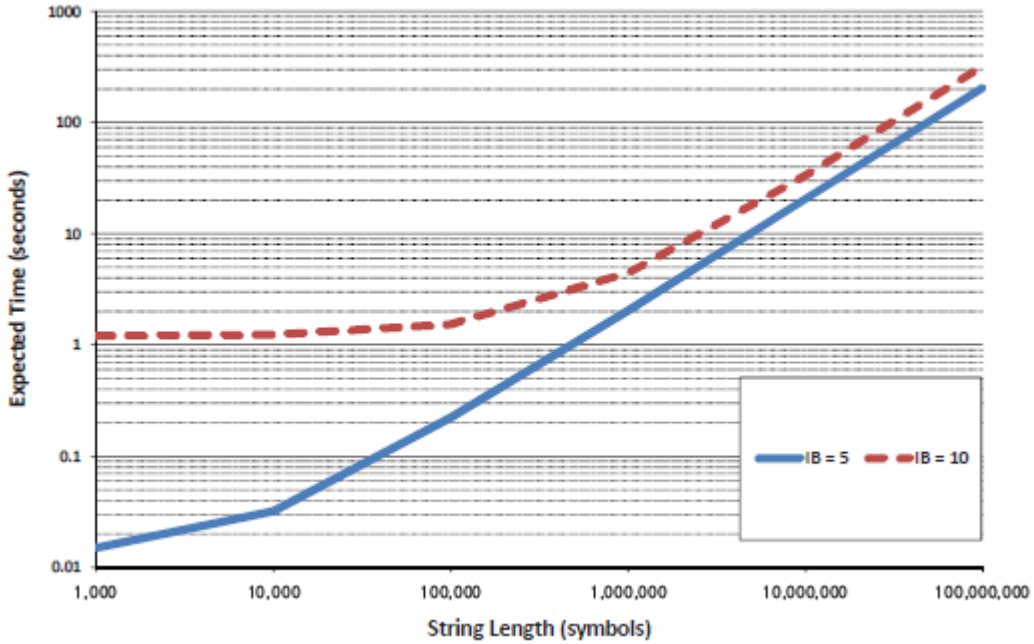
Performance

- Factoring
  - $O(b + L[b^{IB} + (IB - 1)(b^{IB}) + 3N])$
  - $O(b + L[IBb^{IB} + 3N])$
  - $O(L + LN)$

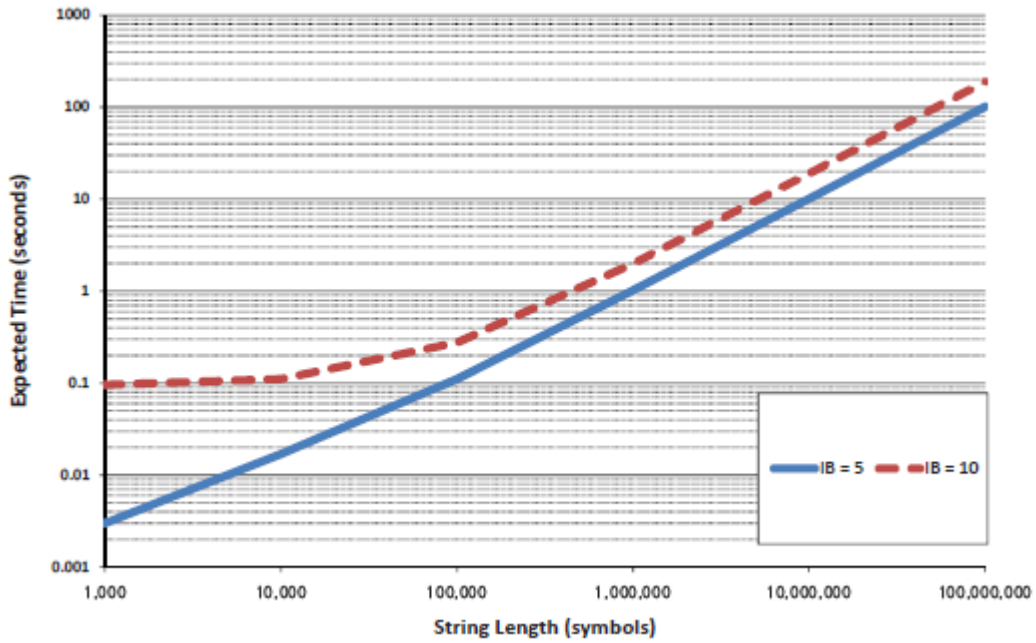
Performance

- Following
  - $O(b + L[R + 2b^{IB} + N])$
  - $O(b + L[3b^{IB} + N])$
  - $O(L + LN)$

Factoring Results



Following Results



Summary

- Factoring provides a means to discover the underlying patterns in a string of genetic data
- Following provides a means to compare any unknown string to one that has been factored previously to determine convergence/divergence
- Both processes have been determined to perform in linear time with respect to the length of the input strings