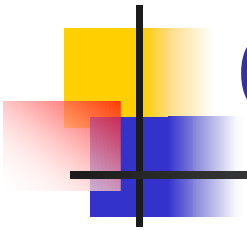
An Implementation of The Teiresias Algorithm

Na Zhao

Chengjun Zhan

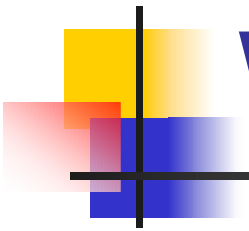


Outline

- Introduction of Pattern Discovery
- Basic Definitions
- Teiresias Algorithm
  - Scan Phase
  - Convolution Phase
- An Example Scenario
- Q & A

What is Pattern Discovery?

- Patterns in proteins:
  - A recurrent region or portion of a protein sequence. It may have a specific structure and it may be functionally significant.
  - Protein family may have similar patterns that can be characterized.
- Pattern Discovery in proteins:
  - Detect patterns from known protein sequences.
  - The result can be used to classify unknown protein sequences.



Why Pattern Discovery Useful?

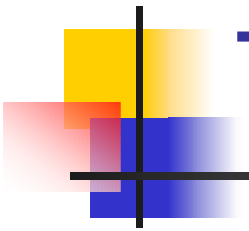
- For some proteins that have similar biological properties on structural or functional features:
  - Group together these protein sequences
  - Discover a set of common sub-sequences
  - Study and observe these sub-sequences
  - The detected patterns may help to classify a protein

Basic Definitions

- ∑ (Basic alphabet set):
  - The amino-acid with names can be abbreviated as the listed symbols in alphabetical order
  - Ex: ∑={A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y}
- . (Wild-card or don't care):
  - a special kind of ambiguous character that matches any character in ∑.
  - Ex: X in protein sequences and N in nucleotide

Basic Definitions

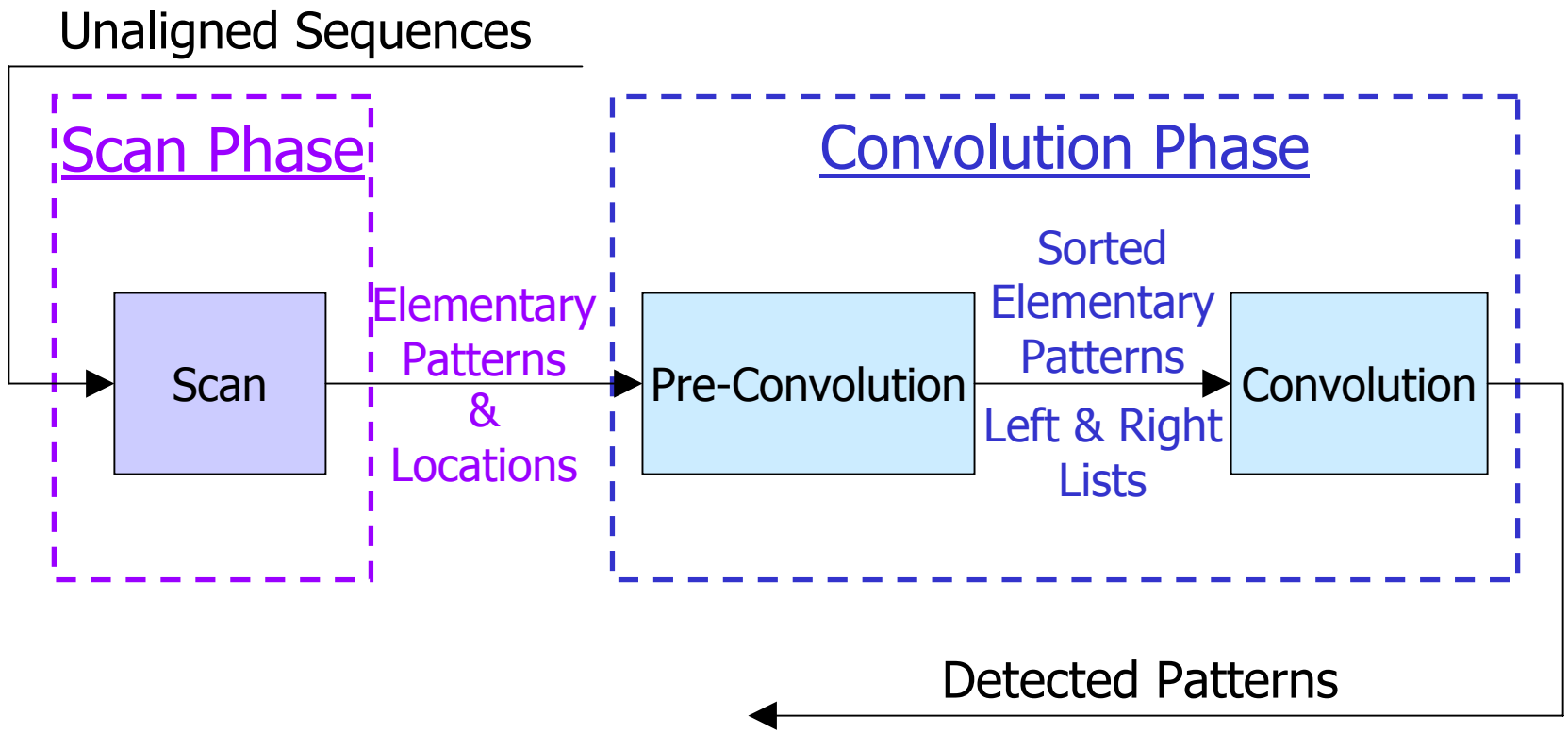
- Pattern P is a (L, W) pattern iff:
  - P is a string of characters (Σ and wild cards '.').
  - P starts and ends with a character from Σ. Characters in Σ are called residues.
  - Any sub pattern of P (i.e subsequence starting and ending with a character from Σ) containing exactly L non-wildcard characters (residues) has length of at most W.
  - Ex. L=3 and W=5, "CD..E" is a (3, 5) pattern.

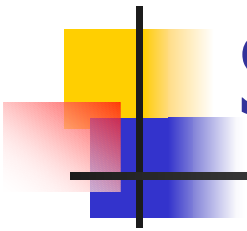


TEIRESIAS Algorithm

- Designed for unaligned sequences
- Basic Idea:
  - If a pattern P is a (L, W) pattern occurring in at least K sequences, then its sub patterns are also (L, W) patterns occurring in at least K sequences (K>=2).
  - Ex: pattern “A.BC” is more specific than “A..C”

Two Phases


Unaligned Sequences
Scan Phase
Scan
Elementary Patterns & Locations
Convolution Phase
Pre-Convolution
Sorted Elementary Patterns
Left & Right Lists
Convolution
Detected Patterns

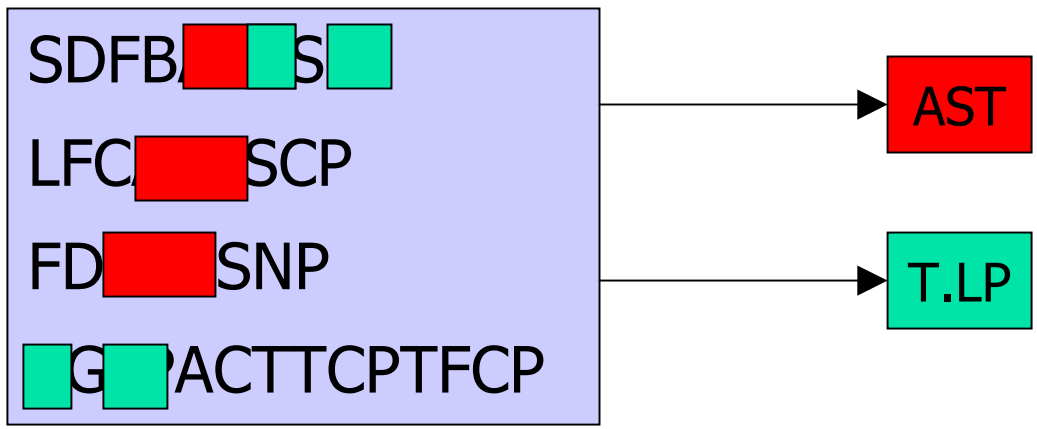


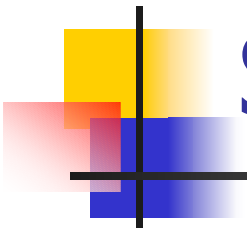
Scan Phase

- Input:
  - Unaligned sequences
  - Parameter: K, L, W
- Output: a set of "Elementary Patterns"
  - Are (L, W) patterns
  - Occur in at least K sequences
  - Contain exactly L non-wildcards

Elementary Pattern Examples

K=2, L=3, W=5

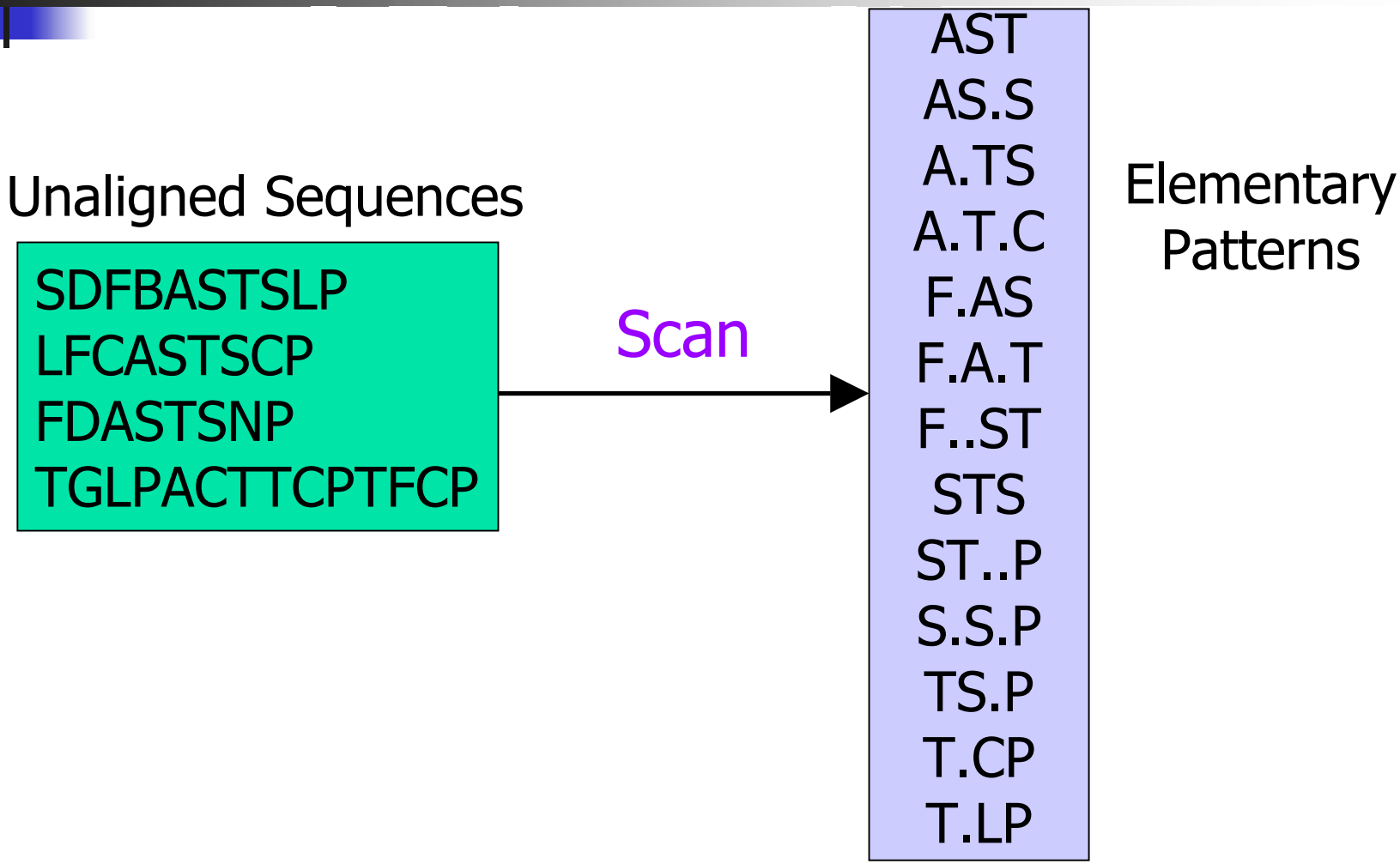


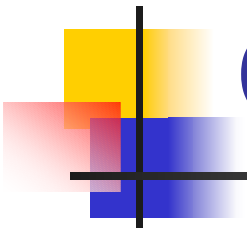


Scan Phase (Cont.)

- Empty the stack of elementary patterns. (EP)
- For each letter in the alphabet, count how many sequences contain this letter.
- If less than K sequences contain this letter, ignore it.
- Otherwise, extend it until it is ignored or it is accepted.
- (Done by Selivonenko & Dyganova)

An example


Unaligned Sequences
SDFBASTSLP
LFCASTSCP
FDASTSNP
TGLPACTTCPTFCP
Scan
AST
AS.S
A.TS
A.T.C
F.AS
F.A.T
F..ST
STS
ST..P
S.S.P
TS.P
T.CP
T.LP
Elementary Patterns



Convolution Phase

- There is a sub-phase named pre-convolution before the convolution phase.
- The goal of the convolution phase is to extend a elementary pattern with other elementary patterns.

Pre-convolution Phase

- Pair-wise < sort the EPs
- Example: suppose some EPs are got from the scan phase,

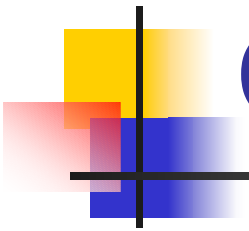
  AA..L AD..G AE..G A.K.G A.L.G A..LG ELA GVS ISR LAD S..SR T.SR VS..T
- After sort, we get,

  ELA GVS ISR LAD AA..L AD..G AE..G VS..T T.SR A.K.G A.L.G A..LG S..SR

Pre-convolution Phase (cont.)

- Related Left and Right vector is constructed.
- Example:

Left	Right
LAD: ELA	ELA: LAD
AD..G: LAD	GVS: VS..T
VS..T: GVS	LAD: AD..G
A..LG: AA..L	AA..L: A..LG



Convolution Phase

- Central idea of convolution phase:
  - Tries to extend the elementary patterns first to the left, if possible.
  - Tries to extend to the right, if possible.
  - Until finally it gets the maximal patterns.
- We will illustrate the whole algorithm by an complete example

An Example Scenario

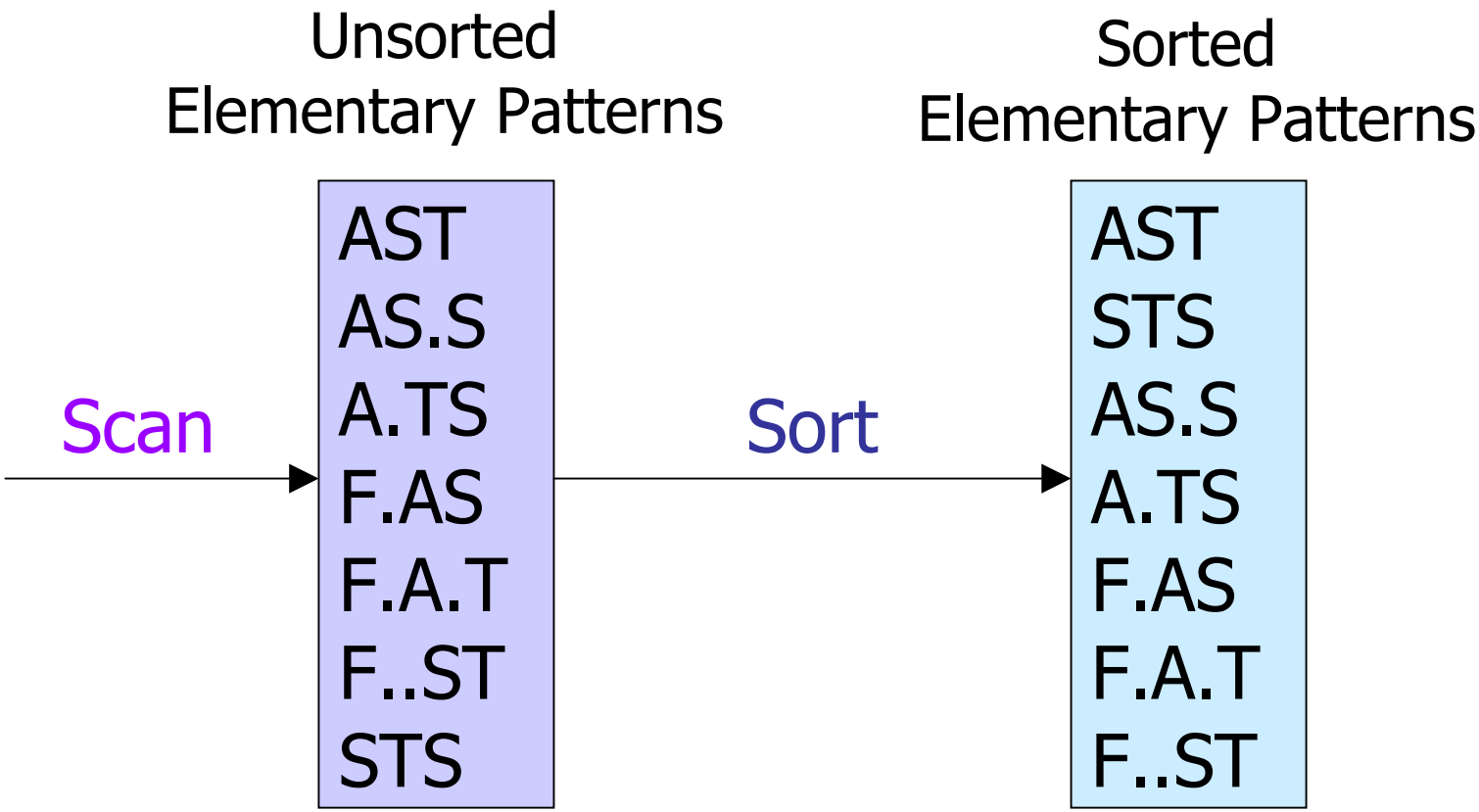
- Suppose there are three sequences:

SDFEASTS
LFCASTS
FDASTSNP

Our goal: to find a maximal pattern, given

K = 2, L = 3, W = 5

An Example Scenario (cont.)

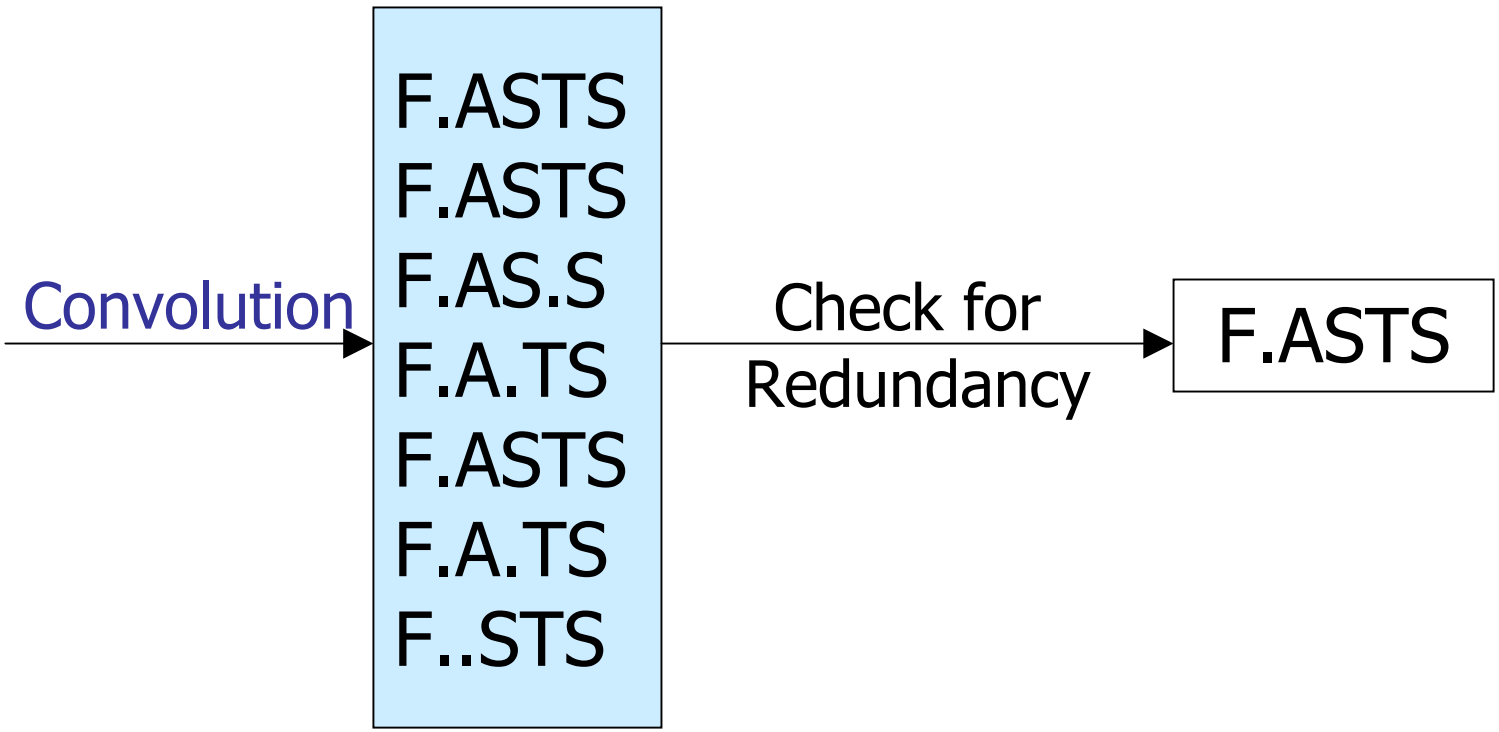


An Example Scenario (cont.)

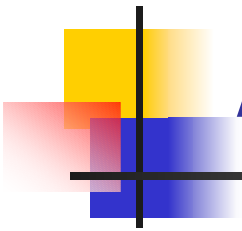
- Left and Right vectors are constructed:

Left	Right
AST: F.AS	AST: STS
STS: AST, F..ST	STS:
AS.S: F.AS	AS.S:
A.TS: F.A.T	A.TS:
F.AS:	F.AS: AST, AS.S
F.A.T:	F.A.T: A.TS
F..ST:	F..ST: STS

An Example Scenario (cont.)



About Our Implementation



- The program is written using

  Visual C++ 6.0

- Command line arguments:

  >Teiresias <Filename> [<K>] [<L>] [<W>]

- We would like to distribute some example results we get by running the program on a large data file

Questions?