Bioinformatics Algorithms

(Fundamental Algorithms, module 2)

Zsuzsanna Lipták

Masters in Medical Bioinformatics
academic year 2018/19, II semester

Strings and Sequences in Computer Science

Some formalism on strings

- Σ a finite set called alphabet

Some formalism on strings

- Σ a finite set called alphabet
- its elements are called characters or letters

Some formalism on strings

- Σ a finite set called alphabet
- its elements are called characters or letters
- $|\Sigma|$ is the size of the alphabet (number of different characters)

Some formalism on strings

- Σ a finite set called alphabet
- its elements are called characters or letters
- $|\Sigma|$ is the size of the alphabet (number of different characters)
- a string over Σ is a finite sequence of characters from Σ

Some formalism on strings

- Σ a finite set called alphabet
- its elements are called characters or letters
- $|\Sigma|$ is the size of the alphabet (number of different characters)
- a string over Σ is a finite sequence of characters from Σ
- we write strings as $s = s_1 s_2 \ldots s_n$ i.e. s_i is the i'th character of s

Some formalism on strings

- Σ a finite set called alphabet
- its elements are called characters or letters
- $|\Sigma|$ is the size of the alphabet (number of different characters)
- a string over Σ is a finite sequence of characters from Σ
- we write strings as $s = s_1 s_2 \ldots s_n$ i.e. s_i is the i'th character of s

N.B.: We number strings from 1, not from 0

Some formalism on strings (cont.)

- $|s|$ is the length of string s

Some formalism on strings (cont.)

- $|s|$ is the length of string s
- ϵ is the empty string, the (unique) string of length 0

Some formalism on strings (cont.)

- $|s|$ is the length of string s
- ϵ is the empty string, the (unique) string of length 0
- Σ^n is the set of strings of length n

Some formalism on strings (cont.)

- $|s|$ is the length of string s
- ϵ is the empty string, the (unique) string of length 0
- Σ^n is the set of strings of length n
- $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$

Some formalism on strings (cont.)

- $|s|$ is the length of string s
- ϵ is the empty string, the (unique) string of length 0
- Σ^n is the set of strings of length n
- $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$ is the set of all strings over Σ

Some formalism on strings: Examples

Examples

- DNA: $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, alphabet size $|\Sigma| = 4$,
  $s = \mathtt{ACCTG}$ is a string of length 5 of Σ, with
  $s_1 = \mathtt{A}$, $s_2 = s_3 = \mathtt{C}$, $s_4 = \mathtt{T}$, $s_5 = \mathtt{G}$.

Some formalism on strings: Examples

Examples

- DNA: $\Sigma = \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$, alphabet size $|\Sigma| = 4$, $s = \mathtt{ACCTG}$ is a string of length 5 of Σ, with $s_1 = \mathtt{A}, s_2 = s_3 = \mathtt{C}, s_4 = \mathtt{T}, s_5 = \mathtt{G}$.
- RNA: $\Sigma = \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{U}\}$, again alphabet size is 4

Some formalism on strings: Examples

Examples

- DNA: $\Sigma = \{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$, alphabet size $|\Sigma| = 4$,
  $s = \texttt{ACCTG}$ is a string of length 5 of Σ, with
  $s_1 = \texttt{A}, s_2 = s_3 = \texttt{C}, s_4 = \texttt{T}, s_5 = \texttt{G}$.
- RNA: $\Sigma = \{\texttt{A},\texttt{C},\texttt{G},\texttt{U}\}$, again alphabet size is 4
- protein: $\Sigma = \{\text{A,C,D,E,F,}\ldots\text{,W,Y}\}$, alphabet size is 20,
  ANRFYWNL is a string over Σ of length 8

Some formalism on strings: Examples

Examples

- DNA: $\Sigma = \{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$, alphabet size $|\Sigma| = 4$,
  $s = \texttt{ACCTG}$ is a string of length 5 of Σ, with
  $s_1 = \texttt{A}, s_2 = s_3 = \texttt{C}, s_4 = \texttt{T}, s_5 = \texttt{G}$.
- RNA: $\Sigma = \{\texttt{A},\texttt{C},\texttt{G},\texttt{U}\}$, again alphabet size is 4
- protein: $\Sigma = \{$A,C,D,E,F,...,W,Y$\}$, alphabet size is 20,
  ANRFYWNL is a string over Σ of length 8
- English alphabet: $\Sigma = \{$a,b,c,...,x,y,z$\}$ of size 26,
  `alphabet` is a string over Σ of length 8

Some formalism on strings

Let $s = s_1 \ldots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s)

Some formalism on strings

Let $s = s_1 \ldots s_n$ be a string over Σ. **ex.** $s = \mathtt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \ldots s_j$ for some $1 \le i \le j \le n$ (i.e., a "contiguous piece" of s) $\mathtt{CCT}, \mathtt{AC}, \ldots$

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s)

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \dots$

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$
  (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$
  (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \dots$
- t is a suffix of s if $t = \epsilon$ or $t = s_i \dots s_n$ for some $1 \leq i \leq n$
  (i.e., an "end" of s)

Some formalism on strings

Let $s = s_1 \ldots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \ldots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \ldots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \ldots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \ldots$
- t is a suffix of s if $t = \epsilon$ or $t = s_i \ldots s_n$ for some $1 \leq i \leq n$ (i.e., an "end" of s) $\texttt{CCTG}, \texttt{G}, \ldots$

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \dots$
- t is a suffix of s if $t = \epsilon$ or $t = s_i \dots s_n$ for some $1 \leq i \leq n$ (i.e., an "end" of s) $\texttt{CCTG}, \texttt{G}, \dots$
- t is a subsequence of s if t can be obtained from s by deleting some (possibly 0, possibly all) characters from s

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \dots$
- t is a suffix of s if $t = \epsilon$ or $t = s_i \dots s_n$ for some $1 \leq i \leq n$ (i.e., an "end" of s) $\texttt{CCTG}, \texttt{G}, \dots$
- t is a subsequence of s if t can be obtained from s by deleting some (possibly 0, possibly all) characters from s $\texttt{AT}, \texttt{CCT}, \dots$

Some formalism on strings

Let $s = s_1 \dots s_n$ be a string over Σ. **ex.** $s = \texttt{ACCTG}$

- t is a substring of s if $t = \epsilon$ or $t = s_i \dots s_j$ for some $1 \leq i \leq j \leq n$ (i.e., a "contiguous piece" of s) $\texttt{CCT}, \texttt{AC}, \dots$
- t is a prefix of s if $t = \epsilon$ or $t = s_1 \dots s_j$ for some $1 \leq j \leq n$ (i.e., a "beginning" of s) $\texttt{AC}, \texttt{ACCTG}, \dots$
- t is a suffix of s if $t = \epsilon$ or $t = s_i \dots s_n$ for some $1 \leq i \leq n$ (i.e., an "end" of s) $\texttt{CCTG}, \texttt{G}, \dots$
- t is a subsequence of s if t can be obtained from s by deleting some (possibly 0, possibly all) characters from s $\texttt{AT}, \texttt{CCT}, \dots$

N.B.

string = sequence, but substring $\neq$ subsequence!

Substrings etc.

N.B.

1. Every substring is a subsequence, but not every subsequence is a substring!

Substrings etc.

N.B.

1. Every substring is a subsequence, but not every subsequence is a substring!
   Ex.: Let $s = \texttt{ACCTG}$, then $\texttt{ACT}$ is a subsequence but not a substring.

Substrings etc.

N.B.

1. Every substring is a subsequence, but not every subsequence is a substring!
   Ex.: Let $s = \texttt{ACCTG}$, then $\texttt{ACT}$ is a subsequence but not a substring.
2. Every prefix and every suffix is a substring.

Substrings etc.

N.B.

1. Every substring is a subsequence, but not every subsequence is a substring!
   Ex.: Let $s = \texttt{ACCTG}$, then $\texttt{ACT}$ is a subsequence but not a substring.
2. Every prefix and every suffix is a substring.
3. t is substring of $s \quad \Leftrightarrow \quad t$ is prefix of a suffix of $s \quad \Leftrightarrow \quad t$ is suffix of a prefix of s

Counting substrings, subsequences etc.

Question

Given $s = s_1 \dots s_n$. How many

- prefixes,
- suffixes,
- substrings,
- subsequences

does s have (exactly, or at most, or at least)?