Bioinformatics Algorithms

Lecture Notes

a.a. 2020/2021

Alberto Mosconi
Politecnico di Milano

Contents

1 Definitions

1.1 Alphabet

Let Σ be a finite set of symbols (alternatively called characters), called the alphabet. No assumption is made about the nature of the symbols.

1.2 String (or word)

A string over Σ is any finite sequence of symbols from Σ. For example if $\Sigma = \{0, 1\}$ then 01011 is a string over Σ.

The **length of a string S is the number of symbols** in s (the length of the sequence) and can be any non-negative integer. It is often denoted as $|S|$.

The **empty string** is the unique string over Σ of length 0, and is denoted as ϵ.

The **set of all strings over Σ of length n** is denoted $\mathbf{\Sigma^n}$. For example , if $\Sigma = 0, 1$, then $\Sigma^2 = \{00, 01, 10, 11\}$. Note that $\Sigma^0 = \{\epsilon\}$ for any alphabet Σ.

The **set of all strings of any length over Σ** is the *Kleene closure* of Σ and is denoted as $\mathbf{\Sigma}^*$. Also:

$$\Sigma^* = \bigcup_{n \in N \cup \{0\}} \Sigma^n$$

The **set of all non-empty strings over Σ** is denoted by $\mathbf{\Sigma}^+$

1.3 Substrings, prefixes and suffixes

A string s is said to be a **substring** (or *factor*) of t if there exist (possibly empty) strings u and v such that $t = usv$.

Given a string t, **suffixes** and **prefixes** are special substrings of t.

A string s is said to be a **prefix** of t if there exists a string u such that $t = su$. If u is nonempty, s is said to be a *proper* prefix of t.

Simmetrically, a string s is said to be a **suffix** of t if there exists a string u such that $t = us$. If u is nonempty, s is said to be a *proper* suffix of t.

1.4 Reverse, palindrome and rotations

The **reverse** of a string is a string with the same symbols but in reverse order. For example, if $s = abc$ (where a, b, and c are symbols of the alphabet), then the reverse of s is cba.

A string that is the reverse of itself is called a **palindrome**, which also includes the empty string and all strings of length 1.

A string $s = uv$ is said to be a **rotation** of t if $t = vu$. For example, if $\Sigma = \{0, 1\}$ the string *0011001* is a rotation of *0100110*, where $u =$ *00110* and $v =$ *01*.

2 Comparing strings

2.1 Hamming distance

The **Hamming distance** between **two strings of equal length** is the **number of positions at which the corresponding symbols are different**.

- "ka*rol*in" and "ka*thr*in" → 3
- 10*1*1*1*01 and 10*0*1*0*01 → 2

With Hamming distance we can formalize *substitutions* in biological sequences - or simply sequencing errors in which the wrong base pair is identified.

2.2 Edit distance

The **edit distance** is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the **minimum number of operations required to transform on string into the other**.

In the *Levenshtein distance* (the most common), edit operations are: **removal**, **insertion**, and **substitution**.

The edit distance between "kitten" and "sitting" is 3. A minimal edit script that transforms the former into the latter is:

- **k**itten → **s**itten (substitute s for k)
- sitt**e**n → sitt**i**n (substitute i for e)
- sittin → sittin**g** (insert g at the end)

The number of solutions (sequences of operations) is infinite.