Description of the program SufPref

0. Introduction
1. Input parameters
2. Probability models
   2.1 Bernoulli model
   2.2 Markov model of order k
   2.3 Hidden Markov model (HHM)
   2.4 Deterministic HHM
3. Pattern description
   3.1 List of words
   3.2 Random generated pattern
   3.3 Given PSSM and cutoff
   3.4 Given PSSM and footprints
   3.5 Word and mismatches
   3.6 Consensus
4. Results

0. Introduction
Given a pattern $H$, i.e. a set of words of the same length $WordLen$ in a given alphabet $A$. Let $T$ be a random text of size $TLen$ generated according to a given probability model. The program supports Bernoulli model, Markov model of an order $k$, Hidden Markov model (HHM). The program computes the probability ($P$-value) to find at least $NOccur$ (possibly overlapping) occurrences of words from a pattern $H$ in a random text $T$.

1. Input parameters
To use the program one has to set the following parameters:
- Text length $TLen$ (integer);
- Number of occurrences $NOccur$ (integer, $NOccur < 1000$);
- Alphabet $A$ (string of the symbols from the alphabet; the order is important);
- Probabilities model (see subsection 2);
- Pattern $H$;

Note, that the probabilities of the symbols from the alphabet have the same order as the symbols.

Optionally you can specify the job title to be shown in the results table. If the title is not specified, the program will generate it. Also you can specify job tag to filter results in summary table.

2. Probability models
2.1 Bernoulli model
Formally, given an alphabet $A = \{a_i\}$ and probabilities $p_{a_i}$ where $\sum p_{a_i} = 1$, the probability to get letter $a$ at any position $j$ is equal to $p_a$ and does not depend on position number nor on letters on previous or subsequent positions.

In this case $|A|$ probabilities have to be given. The probabilities separated by a blank.

Example:

**Probability model:** Bernoulli
**Probability distribution:**
| 0.25 | 0.25 | 0.25 | 0.25 |
2.2 Markov model of order k

In Markov model of order k the letter $X_i$ on position $j$ depends on letters $X_{i-j}, \ldots, X_{i-k}$ at $k$ previous positions. And

$$P(X_1, \ldots, X_n) = \prod_{i=1}^n P(X_i | X_{i-1}, \ldots, X_{i-k})$$

For example, for $k = 2$ and $n = 4$:

$$P(X_1, X_2, X_3, X_4) = P(X_1 | X_2, X_3) \cdot P(X_2 | X_3, X_4) \cdot P(X_3 | X_4) \cdot P(X_4)$$

Thus, to set Markov($k$) model one needs to set all conditional probabilities

$$P(X_i = a_{ij} | X_{i-1} = a_{i-1}, \ldots, X_{i-k} = a_{i-k})$$

for all $(a_{ij}, a_{i-1}, \ldots, a_{i-k}) \in A^{k+1}$

The conditional probabilities are given in a matrix of size $|Alp|^k \times |Alp|$. Rows correspond to $k$ consecutive letters ($k$-grams) arranged in the prefix order. Columns correspond to letters in the alphabet. And an element of the matrix is frequency of a letter occurrence after a $k$-gram.

Also one have to set initial probabilities of all $k$-grams. The probabilities are given in the array $IniProbs$. Using $IniProbs$ we can rewrite above example:

$$P(X_1, X_2, X_3, X_4) = P(X_1 | X_2, X_3) \cdot P(X_2 | X_3, X_4) \cdot P(X_3 | X_4) \cdot IniProbs(X_1, X_2)$$

In the common case initial distribution has to be given. But for asymptotic Markov model initial distribution can be computed as eigen vector of the matrix of conditional probabilities.

The most widely used is Markov model of order 1. The other name is time-homogeneous markov chain. Read about Markov chains in Wikipedia.

Bernoulli model can be described as Markov model of order 0.

Example of completion:

<table>
<thead>
<tr>
<th>Probability model:</th>
<th>Markovian Model</th>
<th>Custom order value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability distribution for Markovian model:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25 0.25 0.25 0.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial probabilities for Markovian model:

| 0.25 0.25 0.25 0.25 |

2.3 Hidden Markov model (HMM)

Let $G = \langle Q, q_0, \pi \rangle$ be an HMM, where $Q$ is the set of states, $q_0 \in Q$ is the initial state; $\pi: Q \times A \times Q \to [0, 1]$ is the probability function. The value $\pi(q, a, q')$ is a probability being in the state $q$ to generate the symbol $a$ and traverse to the state $q'$. The function $\pi$ for any state $q \in Q$ meets the condition:

$$\sum_{a \in A} \sum_{q' \in Q} \pi(q', a, q) = 1$$

The function has to be given by $|Q| \times A$ matrices of the sizes $|Q|$. For each state $q$ one needs the matrix of values $\pi(q, a, q')$ for all $q' \in Q$ and $a \in A$. The matrices have to be separated by line “-ns.” where $ns$ is the number of a current state, $ns = 1, \ldots, |Q| - 1$. 

For example:

**Probability model:** Hidden Markov Model (HMM)

**Probability distribution for HHM:**

```
0.0 0.2 0 0.25 0 0.3 0
0.0 0.0 0.25
-0.1
0.2 0 0 0 0.0 0.5
0.0 0.2 0 0.1
-0.2
0.1 0 0 0 0.5 0.0
0.0 0.4 0
```

### 2.4 Deterministic HHM

The HMM $G$ is called **deterministic** if for any $a \in A$, $q \in Q$ there is only one state $q'$ such that $\pi(q, a, q') > 0$. In this case the function $\pi$ can be described with two functions: (1) **transition** function $\varphi: Q \times A \rightarrow Q$ and (2) **probability** function $\rho: Q \times A \rightarrow [0, 1]$. Namely, in this case $\varphi(q, a)$ is equal to the unique state with $\pi(q, a, q') > 0$ and $\rho(q, a) = \pi(q, a, \varphi(q, a))$. By convention, $\varphi(q, a) = -1$ if there is no transition from a state $q$ with label $a$. Numeration of the states starts with zero.

The functions have to be given by two matrices **Probabilities matrix** and **Transition matrix** of the same sizes $|Q| \times |Alp|$. Rows and columns correspond to states and letters respectively. The matrices have to be given in the following format:

- **probs** (line needed to separate the matrices)
  - **Probabilities matrix**
  - **trans** (line needed to separate the matrices)
  - **Transition matrix**

For example:

**Probability model:** Deterministic HHM

**Probabilities:**

```
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
```

```
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25
```
Transitions:

We can described Markov model of order k as a deterministic HHM. Let M = <Q, □, ρ, ρ0> be a Markov model such that:
1) Q consists of all words of k;
2) for a □ V
   □(q, a) = q' · a, where q' is the suffix of q of length k - 1, if |q| = k;
3) ρ is a probability function.
4) for q □ Q
   IniProbs(q) is initial probability of state q, i.e. probability to start generate a random text with state q.

3. Pattern description

3.1 List of words
In this case all words in the pattern are enumerated.

Example:

Pattern: List of words

Words (newline separated):

Example:

3.2 Random generated pattern
Let number of words NWords in the pattern and pattern length WordLen are given. The program generates NWords words of length WordLen according to the given Bernoulli model. One has to set parameters:
- NWords;
- WordLen;
- Bernoulli probabilities of letters. The probabilities are separated by a blank.

Example:

Number of words: 100  Length of word: 7

Probabilities of letters in the pattern (Bernoulli):

0.25 0.25 0.25 0.25

3.3 Given PSSM and cutoff

The text below was copied from Wikipedia.
A position-specific scoring matrix (PSSM), also called position-specific weight matrix (PSWM) or position weight matrix (PWM), is a commonly used representation of motifs (patterns) in biological sequences.

A PSSM is a matrix of score values that gives a weighted match to any given substring of fixed length. It has one column for each symbol of the alphabet, and one row for each position in the pattern. PSSM score (element \( m_{i,j} \)) is defined as \( \sum_{i=1}^{N} m_{i,j(i)} \), where \( i \) represents position in the substring, \( j(i) \) is the symbol at position \( i \) in the substring, and \( m_{i,j} \) is the score in row \( i \), column \( j \) of the matrix. In other words, a PSSM score is the sum of position-specific scores for each symbol in the substring.

A matrix PSSM contains log odds weights for computing a match. A cutoff is needed to specify whether an input sequence matches the motif or not. A PSSM contains log odds weights for computing a match score. A cutoff is needed to specify whether an input sequence matches the motif or not. Given a PSSM and a cutoff value one can get a set of words scoring above the cutoff.

Example:

**Pattern:**

PSSM and cut-off

**Cut-off:**

3

**PSSM:**

-0.544 0.423 0.356 -0.388
-0.398 0.422 -0.329 0.128
-0.398 -2.054 -2.054 0.992
1.135 -2.054 -1.400 -2.054
1.164 -2.054 -2.054 -2.054

3.4 Given PSSM and footprints.
The information about PSSM was given in the previous subsection. Let given a list of “good” words (footprints) which satisfy some needed conditions. A cutoff value one can take as the minimal score value given by PSSM for each word in the list.

Example:

**Pattern:**

PSSM and footprints

**Footprints:**

AGGTGCCTGGGGTGG
ACGTGG
GGGGGTGTTGAT

3.5 Word and mismatches.
Let a word Motif and a number of mismatches NumMismatches are given. The pattern \( H \) consists of all words which differ from Motif at most in NumMismatches positions. Also the program admits that some positions are fixed (coincide with the same positions in Motif).
Example:

**Pattern:** Word and mismatches

**Word:** ACCTGTG

**Maximum number of mismatches:** 2

**Unchangeable positions in the word (space separated):** 1 2 4

3. **6 Consensus.**

Also a pattern can be presented by a consensus. Let $ConsAlp$ be consensus alphabet where all letters are subsets of the alphabet $A$ and $Consensus$ be a word in $ConsAlp$. Then pattern contain all words in $A$ determined by $Consensus$.

The nomenclature of the International Union of Pure and Applied Chemistry (IUPAC) is as follows:

- $A = \{A\}$ - adenine
- $C = \{C\}$ - cytosine
- $G = \{G\}$ - guanine
- $T = \{T\}$ - thymine
- $U = \{U\}$ uracil
- $R = \{G, A\}$ (purine)
- $Y = \{T, C\}$ (pyrimidine)
- $K = \{G, T\}$ (keto)
- $M = \{A, C\}$ (amino)
- $S = \{G, C\}$ (strong bonds)
- $W = \{A, T\}$ (weak bonds)
- $B = \{G, T, C\}$ (all but A)
- $D = \{G, A, T\}$ (all but C)
- $H = \{A, C, T\}$ (all but G)
- $V = \{G, C, A\}$ (all but T)
- $N = \{A, G, C, T\}$ (any)

In the program letters from $ConsAlp$ except of identical ($X=\{X\}$) has to be given. Identical letters are defined automatically by the program. The format is following:

$X = \{X_1, X_2, \ldots, X_n\}$,

where $X_1, \ldots, X_n \in A$, $X \in ConsAlp$.

Example:

**Pattern:** Consensus

**Consensus alphabet:**

- $Y=\{T, C\}$
- $K=\{G, T\}$
- $N=\{A, C, G, T\}$

**Consensus (a word in consensus alphabet):** AYYTNTK

4. **Results**

The result of the program work is given in the special resulting page. The page appears after entering the input data and pressing the button “Submit”. The calculated P-value and report
about input and some other parameters are given in the section “Your results” in the resulting page. In the section “Your request” of the resulting page the input parameters are duplicated.

*Example of report:*

**Program SufPref**

1. Input parameters
   1.1. Alphabet and probabilities distribution of letters:
   Size of the alphabet: 4
   Alphabet:
   A   C   G   T
   Bernoulli distribution of letters:
   0.28  0.22  0.22  0.28

1.2. Pattern description:
   Input mode: Random words
   Word length: 5
   Number of words: 6

1.3. Parameters for probabilities calculation:
   Size of the text: 100
   Number of occurrences: 1

2. Results
   2.1. Number of overlap classes: 5
   2.2. Number of overlaps: 5
   Number of nodes in Aho-Corasic trie: 29
   2.3. p-Value: 0.005676556649

Here:
A word $w$ is an *overlap* for the pattern $H$ if there exist words $h_1, h_2 \in H$ such as $w$ is the prefix of $h_2$ and $w$ is the suffix of $h_2$. Let $h \in H$ and $l_{pred}(h)$ ($r_{pred}(h)$) is its longest prefix (suffix) which is an overlap. Words $h_1, h_2 \in HH$ are in the same overlap class if $l_{pred}(h_1) = l_{pred}(h_2)$ and $r_{pred}(h_1) = r_{pred}(h_2)$. 