

# **Introdução à Programação de Computadores para Biologia**

## **Estruturas de controle - Loops**

**Aula 07b**

<https://tttorres.github.io/introprog2021/>

# ESTRUTURAS DE CONTROLE

## Repetição

### Problema:

Calcular o tamanho médio de sequências armazenadas em um array.

# ESTRUTURAS DE CONTROLE

## Repetição

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [media.pl](#)
4. Copiar **exemplo04** da página da disciplina.
5. Continuar o script, calculando o tamanho médio das sequências.  
Usar o comando *length()*.

# ESTRUTURAS DE CONTROLE

## Repetição

Script: [media.pl](#)

Calcular média, exemplo04:

```
# calcular a media de tamanho das sequencias em um array

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

# Passo 1: acessar elementos do array, calcular e armazenar tamanhos

$ tamanho = # completar com os comandos;

# Passo 2: calcular a media

$media = # completar com os comandos;

# Passo 3: Imprimir resultado

print "\nExemplo 01:\n";
print "Media de tamanho = $media\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Repetição

Script: [media.pl](#)

Calcular média, exemplo04:

```
# calcular a media de tamanho das sequencias em um array

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

# Passo 1: acessar elementos do array, calcular e armazenar comprimento

$tamanho = length($seqs[0]) + length($seqs[1]) + length($seqs[2]);

# Passo 2: calcular a media

$seqs = @seqs; #numero de elementos do array
$media = $tamanho/$seqs;

# Passo 3: Imprimir resultado

print "\nExemplo 01:\n";
print "Media de tamanho = $media\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Repetição

### Problema:

Calcular o tamanho médio de sequências armazenadas em um array com milhares de elementos.

# ESTRUTURAS DE CONTROLE - LOOPS

**for, foreach, while**

**1. *for***

**2. *foreach***

**3. *while***

# ESTRUTURAS DE CONTROLE - LOOPS

## for, foreach, while

### 1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

### 2. *foreach*

### 3. *while*



# ESTRUTURAS DE CONTROLE - LOOPS

**for, foreach, while**

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [loops.pl](#)
4. Implementar **exemplo05**

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR

Script: [loops.pl](#)

Exemplo 05 (implementar):

```
#!/usr/bin/perl
# script para testar loop com for

# exemplo05

print "\nExemplo 05:\n";

for ($i = 0; $i < 5; $i++) {

    print "$i\n";

}

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR

Script: [loops.pl](#)

Exemplo 06, copiar da página:

```
# for() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

$seqs = @seqs; #numero de elementos do array

print "\nExemplo 06:\n";

print "Numero de sequencias = $seqs\n";

for ( $i = 0; $i < $seqs; $i++ ) {

    print ">Gene$i\n$seqs[$i]\n\n" ;

}

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR

Script: [loops.pl](#)

Exemplo 07, modificar script para calcular a média de tamanho

```
@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");  
$seqs = @seqs; #numero de elementos do array  
  
print "\nExemplo 07:\n";  
  
print "Numero de sequencias = $seqs\n";  
  
for ( $i = 0; $i < $seqs; $i++ ) {  
    # bloco para calcular tamanho  
}  
  
$media = #calcular media  
  
print "Media de tamanho = $media\n";  
  
exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR

Script: [loops.pl](#)

Exemplo 07, modificar script para calcular a média de tamanho

```
@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");  
$seqs = @seqs; #numero de elementos do array  
  
print "\nExemplo 07:\n";  
  
print "Numero de sequencias = $seqs\n";  
  
for ( $i = 0; $i < $seqs; $i++ ) {  
    $tamanho += length($seqs[$i]);  
}  
  
$media = $tamanho/$seqs;  
  
print "Media de tamanho = $media\n";  
  
exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## for, foreach, while

### 1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

### 2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

### 3. *while*

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 08, copiar da página

```
# foreach() para imprimir conteudo de arrays
@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");
###
print "\nExemplo 08:\n";
foreach $seq(@seqs) {
    print "$seq\n\n" ;
}
exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 09, modificar o exemplo08 para imprimir ID sequencial para cada sequencia("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

###

print "\nExemplo 08:\n";

foreach $seq(@seqs) {
    print "$seq\n\n" ;
}

exit;
```



# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 09, modificar o exemplo08 para imprimir ID sequencial para cada sequencia ("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays
print "\nExemplo 09:\n";
$i = 1;
foreach $seq(@seqs) {
    print ">Gene$i\n$seq\n\n";
    ++$i;
}
exit;
```

# ESTRUTURAS DE CONTROLE

## Hashes

### Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimi-las?

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 10, copiar da página

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 10:\n";

#comandos

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 10

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 10:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

# ESTRUTURAS DE CONTROLE

## Hashes

### Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimir as sequências de forma ORDENADA?

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 11, modificar exemplo 10

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 11:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 11, modificar exemplo 10

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 11:\n";

@genes = sort(keys(%seqs));

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

# ESTRUTURAS DE CONTROLE

## Hashes

### Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como encontrar uma sequência específica? Por exemplo, a sequência *GATAGAG* está presente no hash?



# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 12, copiar da página

```
# foreach() para encontrar um valor especifico em hashes  
print "\nExemplo 12:\n";  
$seq_alvo = "GATAGAG"; # esta presente?  
foreach $gene (keys(%seqs)) {  
    #comandos  
}  
exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOREACH

Script: [loops.pl](#)

Exemplo 12

```
# foreach() para encontrar um valor especifico em hashes
print "\nExemplo 12:\n";
$seq_alvo = "GATAGAG"; # esta presente?
foreach $gene (keys(%seqs)) {
    if ($seqs{$gene} eq $seq_alvo) {
        print "Sequencia encontrada: $gene.\n\n";
    }
}
exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## for, foreach, while

### 1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

### 2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

### 3. *while*

```
while (teste) {  
    bloco de comandos  
}
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR vs WHILE

Script: [loops.pl](#)

Exemplo 13 (implementar)

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## FOR vs WHILE

Script: [loops.pl](#)

Exemplo 13 (implementar)

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

# while
$i = 0;

while ($i < 5) {
    print "$i\n";
    ++$i;
}

exit;
```

# ESTRUTURAS DE CONTROLE - LOOPS

## Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 14, copiar da página

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATAGAG");

# for
$seqs = @seqs;

print "\nExemplo 14 (for):\n";

for ( $i = 0; $i < $seqs; $i++ ) {
    print "$seqs[$i]\n" ;
}
```

```
# formas de imprimir arrays
```

```
@seqs = ("ATGGCGTAGATCG","TAAGCCCCGGTATATTTGACCCT", "GATAGAG");
```

```
# for
```

```
$seqs = @seqs;
```

```
print "\nExemplo 14 (for):\n";
```

```
for ( $i = 0; $i < $seqs; $i++ ) {  
    print "$seqs[$i]\n" ;  
}
```

```
# foreach
```

```
print "\nExemplo 14 (foreach):\n";
```

```
foreach $seq (@seqs) {  
    print "$seq\n" ;  
}
```

```
# while
```

```
#$n_seqs = @seqs;
```

```
#$i = 0;
```

```
#print "\nExemplo 14 (while):\n";
```

```
#while ($i < $n_seqs) {  
#    print "$seqs[$i]\n";  
#}
```

# ESTRUTURAS DE CONTROLE

**for, foreach, while**

**Exercício:**

Usando loop em Perl, crie um array com os números de 10 a 900, contando de 10 em 10.