

Introdução à Programação de Computadores para Biologia

Módulos, CPAN e Bioperl

Aula 14

<https://tttorres.github.io/introprog2024/>

MÓDULOS

SeqAleatoria.pl

```
#!/usr/bin/perl/

## Script para exemplificar subrotinas

use strict;
use aula13;

## Declaracao de variaveis
my $tamanho; my $sequencia; my $revSeq;

# criar a sequencia aleatoria
$tamanho = 200;
$sequencia = seqAleatoria($tamanho);

# inverter a sequencia
$revSeq = reverse($sequencia);

# gerar o complemento reverso
$revSeq =~ tr/ATCG/atcg/;

# imprimir (fasta)
print ">seq\n";
for (my $i = 0; $i < $tamanho; $i+=70) {
    print substr($revSeq,$i,70), "\n";
}

exit;
```

MÓDULOS

aula13.pm

```
##  Modulo com as minhas subrotinas

##  modulo aula13

sub seqAleatoria {

    $seq = "";
    $tamanho = shift || 40;
    for ($i = 0; $i < $tamanho; $i++) {
        $base = int(rand(4));
        if ($base == 0) { $seq .= "A"; next; }
        if ($base == 1) { $seq .= "T"; next; }
        if ($base == 2) { $seq .= "C"; next; }
        if ($base == 3) { $seq .= "G"; next; }
    }
    return $seq;
}

sub soma {
    ($n1, $n2) = @_;
    $valor = $n1 + $n2;
    return $valor;
}

1;
```

MÓDULOS

Módulos incluídos na distribuição do Perl

- strict
- warnings
- Getopt::Std
- GetOpt::Long
- Time::Local
- File::Basename
- DBI
- CGI
- Net::FTP

Outros módulos incluídos:

<http://perldoc.perl.org/index-modules-A.html>

MÓDULOS

Módulos incluídos na distribuição do Perl

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [modulos.pl](#)
4. Copiar **exemplo01** da página da disciplina.
5. Qual é a função do script?

MÓDULOS

Módulos incluídos na distribuição do Perl

Script: [modulos.pl](#)

```
#!/usr/bin/perl

# qual a funcao deste script?

$seq = "atg att gaa cca tga";

$codons = count_codons($seq); # qual a funcao desta linha?

print "A sequencia indicada: \"$seq\" contem $codons codons\n";

exit;

sub count_codons {
    $seq = shift;
    $seq = uc($seq);          # qual a funcao desta linha?
    $seq =~ s/\s+//g;         # e desta?
    $codons = length($seq) / 3;
    return($codons);
}
```

MÓDULOS

Módulos incluídos na distribuição do Perl

1. *Strict*

```
#!/usr/bin/perl

use strict;

$seq = "atg att gaa cca tga";

$codons = count_codons($seq); # qual a funcao desta linha?

print "A sequencia indicada: \"$seq\" contem $codons codons\n";

exit;

sub count_codons {
    $seq = shift;
    $seq = uc($seq);          # qual a funcao desta linha?
    $seq =~ s/\s+//g;         # e desta?
    $codons = length($seq) / 3;
    return($codons);
}
```

MÓDULOS

Módulos incluídos na distribuição do Perl

1. *Strict*

```
#!/usr/bin/perl
use strict;

my $seq; my $codons;

$seq = "atg att gaa cca tga";
$codons = count_codons($seq); # qual a funcao desta linha?
print "A sequencia indicada: \"$seq\" contem $codons codons\n";

exit;

sub count_codons {
    my $seq; my $codons;
    $seq = shift;
    $seq = uc($seq);           # qual a funcao desta linha?
    $seq =~ s/\s+//g;          # e desta?
    $codons = length($seq) / 3;
    return($codons);
}
```


MÓDULOS

Módulos incluídos na distribuição do Perl

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [warnings.pl](#)
4. Copiar **exemplo02** da página da disciplina.
5. Qual é a função do script?

MÓDULOS

Módulos incluídos na distribuição do Perl

Script: [warnings.pl](#)

```
#!/usr/bin/perl

use strict;

my ($x, $y, $z) = (1, 2, 3, 4, 5);
print "x=$x y=$y z=$z\n";

my ($a, $b, $c) = (1, 2);
print "Conteúdo de c=$c\n";
print "Comprimento de c=", length($c), "\n";
print "Resultado de c + a = ", $a + $c, "\n";

exit;
```

MÓDULOS

Módulos incluídos na distribuição do Perl

Script: [warnings.pl](#), terminal:

```
Darwin:Introprog Tatiana$ perl warnings.pl  
x=1 y=2 z=3
```

```
Conteudo de c =  
Comprimento de c =  
Resultado de c + a = 1
```

MÓDULOS

Módulos incluídos na distribuição do Perl

2. *Warnings*

```
#!/usr/bin/perl

use strict;
use warnings;

my ($x, $y, $z) = (1, 2, 3, 4, 5);
print "x=$x y=$y z=$z\n";

my ($a, $b, $c) = (1, 2);
print "Conteúdo de c=$c\n";
print "Comprimento de c=", length($c), "\n";
print "Resultado de c + a = ", $a + $c, "\n";

exit;
```

MÓDULOS

Módulos incluídos na distribuição do Perl

2. *Warnings*; no terminal:

```
Darwin:Introprog Tatiana$ perl warnings.pl
x=1 y=2 z=3

Use of uninitialized value $c in concatenation (.) or string at
warnings.pl line 12.
Conteudo de c=
Use of uninitialized value in print at warnings.pl line 13.
Comprimento de c=
Use of uninitialized value $c in addition (+) at warnings.pl line 14.
Resultado de c + a = 1
```

MÓDULOS

Módulos incluídos na distribuição do Perl

Problema

Script com muitos argumentos. Por exemplo: abrir um arquivo com sequências, blastar todas as sequências contra um arquivo de banco de dados e selecionar os melhores hits de acordo com os parâmetros pré-definidos.

MÓDULOS

Módulos incluídos na distribuição do Perl

ARGUMENTOS PARA RODAR O BLAST:

- Número de processadores a serem utilizados
- Nome do programa (blastn, blastx ...)
- Arquivo de entrada (sequências)
- Banco de dados (sequências)

ARGUMENTOS PARA SELECIONAR OS HITS A PARTIR DA SAÍDA DO BLAST:

- E-value
- Identidade mínima
- Tamanho mínimo

MÓDULOS

Módulos incluídos na distribuição do Perl

Exemplo:

```
Darwin:~ Tatiana$ perl blastLocal.pl 12 blastn sequencias  
db.fasta 10e-4 70 50
```

- Número de processadores a serem utilizados
- Nome do programa (blastn, blastx ...)
- Arquivo de entrada (sequências)
- Banco de dados (sequências)
- E-value
- Identidade mínima
- Tamanho mínimo

MÓDULOS

Módulos incluídos na distribuição do Perl

Exemplo:

```
Darwin:~ Tatiana$ perl blastLocal.pl 12 blastn sequencias
db.fasta 10e-4 70 50
```

```
#!/usr/bin/perl/
```

```
use strict;
```

```
## Pegando os argumentos do @ARGV
```

```
my $threads    = $ARGV[0]; # numero de processadores
my $program    = $ARGV[1]; # nome do programa
my $input      = $ARGV[2]; # arquivo de entrada
my $database    = $ARGV[3]; # banco de dados
my $evalue     = $ARGV[4]; # e-value
my $identity    = $ARGV[5]; # identidade mínima
my $length     = $ARGV[6]; # tamanho mínimo
```

MÓDULOS

Módulos incluídos na distribuição do Perl

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [help.pl](#)
4. Copiar **exemplo03** da página da disciplina.
5. Escrever uma subrotina que busca por argumentos de linha de comando com os seguintes padrões: -help e -h. Caso o usuário entre com algum desses argumentos, a subrotina deve retornar um valor verdadeiro. Nesse caso, o programa deve imprimir a mensagem de ajuda por meio da variável \$USAGE e terminar.

MÓDULOS

Módulos incluídos na distribuição do Perl

Script: [help.pl](#)

```
#!/usr/bin/perl/

use strict; use warnings;

my $help = argvtest(\@ARGV);

my $USAGE = "Usage: perl $0 help.pl <input> <output>\n";

if ($help) {
    die $USAGE;
}

my $input  = $ARGV[0];
my $output = $ARGV[1];

open(SEQ, "<$input") || die "Nao foi possivel abrir o arquivo $input\n";
open(OUT, ">$output") || die "Nao foi possivel abrir o arquivo $output\n";

while (<SEQ>) {
    print OUT;
}

close SEQ;

exit;
```

MÓDULOS

Módulos incluídos na distribuição do Perl

Script: [help.pl](#)

```
# subrotina para ajuda

sub argvtest {

    my $arrRef = shift;
    my @array = @{$arrRef};

    foreach my $arg (@array) {
        if (($arg eq "-h") or ($arg eq "-help")){
            return 1;
        }
    }
}
```

MÓDULOS

```
Darwin:~ Tatiana$ perl blastLocal.pl 12 blastn sequencias.fasta  
db.fasta 10e-4 70 50
```

ARGUMENTOS PARA RODAR O BLAST:

- Número de processadores a serem utilizados = -t
- Nome do programa (blastn, blastx ...) = -p
- Arquivo de entrada (sequências) = -s
- Banco de dados (sequências) = -d

ARGUMENTOS PARA SELECIONAR OS HITS A PARTIR DA SAÍDA DO BLAST:

- E-value = -e
- Identidade mínima = -i
- Tamanho mínimo = -l
- Ajuda = -h

MÓDULOS

Módulos incluídos na distribuição do Perl

Exemplo:

```
Darwin:~ Tatiana$ perl blastLocal.pl -t 12 -p blastn -s  
sequencias.fasta -d db.fasta -e 10e-4 -i 70 -l 50
```

MÓDULOS

Módulos incluídos na distribuição do Perl

3. *Getopt*

```
Darwin:~ Tatiana$ perl blastLocal.pl -t 12 -p blastn -s  
sequencias.fasta -d db.fasta -e 10e-4 -i 70 -l 50
```

```
#!/usr/bin/perl/
```

```
use strict;  
use Getopt::Std;
```

```
## Pegando os argumentos com o módulo Getopt
```

```
my %opts = ();  
getopts('t:s:p:d:e:i:l:h', \%opts);
```

MÓDULOS

Módulos incluídos na distribuição do Perl

3. *Getopt*

```
#!/usr/bin/perl/  
  
use strict;  
use Getopt::Std;  
  
## Pegando os argumentos com o módulo Getopt  
my %opts = ();  
getopts('t:s:p:d:e:i:l:h', \%opts);
```

getopts -> Subrotina do pacote Std do módulo *Getopt*

Getopt

```
#!/usr/bin/perl/

use strict;
use Getopt::Std;

## Pegando os argumentos com o módulo Getopt
my %opts = ();
getopts('t:s:p:d:e:i:l:h', \%opts);
```

't:s:p:d:e:i:l:h' -> argumentos

- t = número de processadores
- s = Arquivo de entrada (sequencias)
- p = Nome do programa (blastn, blastx ...)
- d = Banco de dados (sequencias)
- e = E-value
- i = Identidade mínima
- l = Tamanho mínimo

MÓDULOS

Módulos incluídos na distribuição do Perl

3. *Getopt*

```
#!/usr/bin/perl/  
  
use strict;  
use Getopt::Std;  
  
# Capturando os argumentos com o módulo Getopt  
my %opts = ();  
getopts('t:s:p:d:e:i:l:h', \%opts);
```

%opts -> Referência para o hash %opts

MÓDULOS

Módulos incluídos na distribuição do Perl

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [opts.pl](#)
4. Copiar **exemplo04** da página da disciplina.

Getopt

```
#!/usr/bin/perl/
use strict;
use Getopt::Std;

# Capturando os argumentos com o módulo Getopt
my %opts = ();
getopts('t:s:p:d:e:i:l:h', \%opts);

# Uso do programa
my ($USAGE) = "\nUSAGE: $0\n".
    "\t\t-t Numero de processadores \n".
    "\t\t-s Entrada - sequencias em formato fasta (".
    "\t\t-d Banco de sequencias (indispensavel)\n".
    "\t\t-p Nome do programa\n".
    "\t\t-e E-value\n".
    "\t\t-i % minima de identidade\n".
    "\t\t-l Tamanho mínimo do alinhamento\n\n";

# Mostra a ajuda e encerra
if ($opts{h}) {
    die $USAGE;
}
```

MÓDULOS

Módulos incluídos na distribuição do Perl

3. *Getopt*

```
# Atribui os argumentos a variáveis
my $threads    = $opts{t} || 1;
my $input_seq  = $opts{s} || die $USAGE;
my $db         = $opts{d} || die $USAGE;
my $prog       = $opts{p} || 'blastn';
my $eval       = $opts{e} || 0.0001;
my $id         = $opts{i} || 70;
my $len        = $opts{l} || 200;
```

MÓDULOS

Módulos incluídos na distribuição do Perl

4. Time::Local

Funções para trabalhar com datas

5. File::Path

Criar e manipular diretórios dentro de um script perl

6. File::Basename

Separar nomes de arquivos em diretório, nome e sufixo

Outros módulos incluídos: CGI, DBI, Net::FTP

<http://perldoc.perl.org/index-modules-A.html>

MÓDULOS

Módulos incluídos na distribuição do Perl

Como saber se um módulo está instalado?

```
#!/usr/bin/perl/  
  
use strict;  
use Getopt::Std;  
  
# Capturando os argumentos com o módulo Getopt  
my %opts = ();  
getopts('t:s:p:d:e:i:l:h', \%opts);
```

MÓDULOS

Módulos incluídos na distribuição do Perl

Como saber se um módulo está instalado?

```
#!/usr/bin/perl/  
  
use strict;  
use Date::Simple;  
  
## Capturando os argumentos com o módulo Getopt  
my %opts = ();  
getopts('t:s:p:d:e:i:l:h', \%opts);
```


MÓDULOS

Perl one-liners

No terminal

```
Darwin:~ Tatiana$ perl -e 'print "Hello world!\n"'
```

MÓDULOS

Perl one-liners

No terminal

```
Darwin:~ Tatiana$ perl -e 'print "Hello world!\n"'  
Darwin:~ Tatiana$ perl -h
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

MÓDULOS

Perl one-liners

No terminal

```
Darwin:~ Tatiana$ perl -e 'print "Hello world!\n"'  
Darwin:~ Tatiana$ perl -mGetopt::Std -e 'print "Modulo  
presente\n"'
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

MÓDULOS

Perl one-liners

No terminal

```
Darwin:~ Tatiana$ perl -e 'print "Hello world!\n"'
Darwin:~ Tatiana$ perl -mGetopt::Std -e 'print "Modulo presente\n"'
Darwin:~ Tatiana$ perl -mSpreadsheet::WriteExcel -e
'print "Modulo presente\n"'
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

MÓDULOS

Perl one-liners

No terminal

```
Darwin:~ Tatiana$ perl -e 'print "Hello world!\n"'
Darwin:~ Tatiana$ perl -mGetopt::Std -e 'print "Modulo presente\n"'
Darwin:~ Tatiana$ perl -mSpreadsheet::WriteExcel -e 'print "Modulo pr
Can't locate Spreadsheet/WriteExcel.pm in @INC (you may need to insta
Spreadsheet::WriteExcel module) (@INC contains: /Library/Perl/5.18/darw
hread-multi-2level /Library/Perl/5.18 /Network/Library/Perl/5.18/darw
ead-multi-2level /Network/Library/Perl/5.18 /Library/Perl/Updates/5.1
rwin-thread-multi-2level /Library/Perl/Updates/5.18.2 /System/Library
/5.18/darwin-thread-multi-2level /System/Library/Perl/5.18 /System/Li
Perl/Extras/5.18/darwin-thread-multi-2level /System/Library/Perl/Extr
8 .).
BEGIN failed--compilation aborted.
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

MÓDULOS

Módulos NÃO incluídos na distribuição do Perl

- Date::Simple
- File::Slurp
- Mail::Sendmail
- Spreadsheet::ParseExcel
- Spreadsheet::WriteExcel
- GD::Graph
- BioPerl

CPAN

Comprehensive Perl Archive Network



Comprehensive Perl Archive Network

YOU CAN NEVER HAVE TOO MANY PERL MODULES

[Home](#)[Modules](#)[Ports](#)[Perl Source](#)[FAQ](#)[Mirrors](#)

Search:

Welcome to CPAN

The Comprehensive Perl Archive Network (CPAN) currently has [176,622 Perl modules](#) in 39,055 distributions, written by 13,633 authors, [mirrored](#) on 261 servers.

The archive has been online since October 1995 and is constantly growing.

Search CPAN via

- [metacpan.org](#)

Recent Uploads

- [Tie-Hash-RedisDB-1.03](#)
- [Tk-PerlMethodList-0.08](#)
- [Tk-PerlInheritanceTree-0.06](#)
- [Catalyst-Runtime-5.90120](#)
- [Git-MoreHooks-0.008](#)
- [Net-Async-Redis-1.012](#)
- [Sparrow-0.3.2](#)
- [XML-Invisible-0.04](#)
- [Mojolicious-Plugin-FastHelpers-0.02](#)
- [Mojolicious-Plugin-TemplateToolkit-0.005](#)
- [more...](#)

Getting Started

- [Installing Perl Modules](#)
- [Learn Perl](#)

How to contribute

- [Read this](#)
- Visit <https://pause.perl.org/>

Perl Resources

- [The Perl Programming language](#)
- [Perl Documentation](#)
- [Mailing Lists](#)
- [Perl FAQ](#)
- [Scripts Repository](#)

Yours Eclectically, The Self-Appointed Master Librarians (OOK!) of the CPAN.
© 1995-2010 Jarkko Hietaniemi. © 2011-2017 [Perl.org](#). All rights reserved. [Disclaimer](#).

Master mirror hosted by  and 

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo [CPAN.pm](#)

No terminal (entrando no modo interativo):

```
Darwin:~ Tatiana$ perl -MCPAN -e shell
```


CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo [CPAN.pm](#)

Se for a primeira utilização do CPAN:

```
Are you ready for manual configuration? [yes]
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Se **não** for a primeira utilização do CPAN:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell  
  
cpan shell -- CPAN exploration and modules installation (  
Enter 'h' for help.  
  
cpan[1]>
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Se **não** for a primeira utilização do CPAN:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell  
  
cpan shell -- CPAN exploration and modules installation (  
Enter 'h' for help.  
  
cpan[1]> h
```

CPAN

Comprehensive Perl Archive Network

```
cpan[1]> h
```

```
Display Information (ver 1.9456)
```

command	argument	description
a,b,d,m	WORD or /REGEXP/	about authors, bundles, distributions, modules
i	WORD or /REGEXP/	about any of the above
ls	AUTHOR or GLOB	about files in the author's directory

(with WORD being a module, bundle or author name or a distribution name of the form AUTHOR/DISTRIBUTION)

```
Download, Test, Make, Install...
```

get	download	clean	make clean
make	make (implies get)	look	open subshell in dist directory
test	make test (implies make)	readme	display these README files
install	make install (implies test)	perldoc	display POD documentation

```
Upgrade
```

r	WORDS or /REGEXP/ or NONE	report updates for some/matching/all modules
upgrade	WORDS or /REGEXP/ or NONE	upgrade some/matching/all modules

```
Pragmas
```

force	CMD	try hard to do command	fforce	CMD	try harder
notest	CMD	skip testing			

```
...
```

```
cpan[2]>
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Instalação rápida:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell

cpan shell -- CPAN exploration and modules installation (v1.2.0)
Enter 'h' for help.

cpan[1]> h

cpan[2]> o conf prerequisites_policy follow
```

automatically ('follow'), ask you for confirmation ('ask'), or just ignore them ('ignore')

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Instalação rápida:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell
```

```
cpan shell -- CPAN exploration and modules installation (v1.0.2)
Enter 'h' for help.
```

```
cpan[1]> h
```

```
cpan[2]> o conf prerequisites_policy follow
prerequisites_policy [follow]
```

```
Please use 'o conf commit' to make the config permanent!
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Instalação rápida:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell

cpan shell -- CPAN exploration and modules installation (v1.2.0)
Enter 'h' for help.

cpan[1]> h

cpan[2]> o conf prerequisites_policy follow
prerequisites_policy [follow]
Please use 'o conf commit' to make the config permanent!
cpan[3]> o conf commit
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Instalação rápida:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell
```

```
cpan[1]> h
```

```
cpan[2]> o conf prerequisites_policy follow  
prerequisites_policy [follow]
```

```
Please use 'o conf commit' to make the config permanent!
```

```
cpan[3]> o conf commit
```

```
commit: wrote '/Users/Tatiana/.cpan/CPAN/MyConfig.pm'
```


CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Instalação rápida:

```
Darwin:~ Tatiana$ perl -MCPAN -e shell
```

```
cpan[1]> h
```

```
cpan[2]> o conf prerequisites_policy follow  
prerequisites_policy [follow]
```

```
Please use 'o conf commit' to make the config permanent!
```

```
cpan[3]> o conf commit
```

```
commit: wrote '/Users/Tatiana/.cpan/CPAN/MyConfig.pm'
```

```
cpan[4]> install File::Slurp
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Se não houver permissão para instalação:

```
Darwin:~ Tatiana$ cpan[4]> q
```

```
Darwin:~ Tatiana$ sudo perl -MCPAN -e shell
```

CPAN

Comprehensive Perl Archive Network

Instalação interativa de módulos do CPAN utilizando o módulo **CPAN.pm**

Se não houver permissão para instalação:

```
Darwin:~ Tatiana$ sudo perl -MCPAN -e shell
```

```
cpan[1]> o conf
```

prerequisites_policy [follow]?

```
cpan[2]> install File::Slurp
```

MÓDULOS

Perl one-liners

No terminal, verificar a instalação do módulo `File::Slurp`

```
Darwin:~ Tatiana$ perl -mFile::Slurp -e 'print  
"Modulo presente\n"'
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

MÓDULOS

Módulos instalados

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [slurp.pl](#)
4. Copiar **exemplo05** da página da disciplina.

MÓDULOS

Módulos instalados

4. *File::Slurp*

```
use strict;
use warnings;
use File::Slurp;

my $file = shift @ARGV;
my $conteudo = read_file($file);

# Criar um array de letras de A a T
my @letras = ('A'..'T');

# Substituir todas as ocorrências do padrão anterior pelo
$conteudo =~ s/Grupo (\d+)/'Grupo '.$letras[$1-1]/ge;

# Gravar o conteúdo modificado de volta ao arquivo
write_file($file, $conteudo);
```

MÓDULOS

Módulos instalados

4. *File::Slurp*

No terminal:

```
Darwin:~ Tatiana$ cat dados.txt
Grupo 1: 2.3, 3.4, 5.6, 7.1, 6.2
Grupo 2: 4.1, 2.8, 3.7, 6.5, 5.9
Grupo 3: 1.2, 2.3, 3.4, 1.9, 2.5
Grupo 4: 3.2, 4.1, 5.5, 6.6, 4.3
Grupo 5: 2.8, 3.0, 4.1, 5.2, 6.8
...
```

```
Darwin:~ Tatiana$ perl slurp.pl dados.txt
```

MÓDULOS

Módulos instalados

4. File::Slurp

No terminal:

```
Darwin:~ Tatiana$ cat dados.txt
Grupo A: 2.3, 3.4, 5.6, 7.1, 6.2
Grupo B: 4.1, 2.8, 3.7, 6.5, 5.9
Grupo C: 1.2, 2.3, 3.4, 1.9, 2.5
Grupo D: 3.2, 4.1, 5.5, 6.6, 4.3
Grupo E: 2.8, 3.0, 4.1, 5.2, 6.8
...
```


BIOPERL

Ferramentas (pacotes) em Perl para bioinformática

BIOPERL

Ferramentas (pacotes) em Perl para bioinformática

Conjunto de pacotes com código aberto para lidar com problemas comuns em Bioinformática, Genômica e outros problemas biológicos




Exemplos:


- Ler e escrever sequências em vários formatos (Fasta, Fastq, GenBank, SwissProt ...)
- Extrair anotações de genes de arquivos como GenBank, SwissProt ...
- Ler e analisar resultados do BLAST.
- Ler e processar árvores filogenéticas e alinhamentos múltiplos de sequências.
- Analisar dados de SNP.
- ...

BIOPERL


<http://bio.perl.org/>

BioPerl







The BioPerl Project is an international association of users & developers of open source Perl tools for bioinformatics, genomics and life science




Installation
Installing the current version




Documentation
HOWTOs and Scrapbook code




Support
BioPerl Mailing Lists



Issues
Submit bugs or enhancement requests to GitHub




Code
BioPerl Packages at GitHub



OBF
The Open Bioinformatics Foundation

© BioPerl 2016

If you are interested in contributing please contact us at [bioperl-l](mailto:bioperl-l@lists.sourceforge.net) 



Main Links

[Main Page](#)
[Getting Started](#)
[Downloads](#)
[Installation](#)
[Recent changes](#)
[Random page](#)

documentation

[Quick Start](#)
[FAQ](#)
[HOWTOs](#)
[API Docs](#)
[Scrapbook](#)
[Tutorials](#)
[Deobfuscator](#)
[Browse Modules](#)

community

[News](#)
[Mailing lists](#)
[Supporting BioPerl](#)
[BioPerl Media](#)
[Hot Topics](#)
[About this site](#)
[News](#)
[Mailing lists](#)
[Supporting BioPerl](#)
[BioPerl Media](#)
[Hot Topics](#)
[About this site](#)

development

[Developer Information](#)
[Advanced BioPerl](#)
[Git](#)
[Module Tests](#)
[Bugs](#)

Toolbox

[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)

Category [Discussion](#)

[Read](#) [View source](#) [View history](#)

Category:Modules

All BioPerl modules.

Pages in category "Modules"

The following 200 pages are in this category, out of 1,525 total.
(previous 200) (next 200)

B

- [Bio::Tools::Run::Samtools::Config](#)

M

- [Module Discussion:Bio::Assembly::Contig](#)
- [Module Discussion:Bio::Assembly::ContigAnalysis](#)
- [Module Discussion:Bio::Assembly::IO](#)
- [Module Discussion:Bio::Assembly::IO::phrap](#)
- [Module Discussion:Bio::Assembly::IO::tigr](#)
- [Module Discussion:Bio::Assembly::Scaffold](#)
- [Module Discussion:Bio::Assembly::Singlet](#)
- [Module Discussion:Bio::Assembly::Tools::ContigSpectrum](#)
- [Module:Bio::Align::Alignl](#)
- [Module:Bio::Align::DNAStatistics](#)
- [Module:Bio::Align::PairwiseStatistics](#)
- [Module:Bio::Align::ProteinStatistics](#)
- [Module:Bio::Align::Statisticsl](#)
- [Module:Bio::Align::Utilities](#)
- [Module:Bio::AlignO](#)
- [Module:Bio::AlignO::arp](#)
- [Module:Bio::AlignO::bl2seq](#)
- [Module:Bio::AlignO::clustalw](#)
- [Module:Bio::AlignO::emboss](#)
- [Module:Bio::AlignO::fasta](#)
- [Module:Bio::AlignO::largemultifasta](#)
- [Module:Bio::AlignO::maf](#)
- [Module:Bio::AlignO::mase](#)
- [Module:Bio::AlignO::mega](#)
- [Module:Bio::AlignO::meme](#)
- [Module:Bio::AlignO::metafasta](#)
- [Module:Bio::AlignO::msf](#)
- [Module:Bio::AlignO::nexml](#)
- [Module:Bio::AlignO::nexus](#)
- [Module:Bio::AlignO::pfam](#)
- [Module:Bio::AlignO::phylip](#)
- [Module:Bio::AlignO::po](#)
- [Module:Bio::AlignO::proda](#)
- [Module:Bio::AlignO::prodorm](#)
- [Module:Bio::AlignO::psi](#)
- [Module:Bio::AlignO::selex](#)
- [Module:Bio::AlignO::stockholm](#)
- [Module:Bio::AlignO::xmfa](#)
- [Module:Bio::Analysisl](#)
- [Module:Bio::Analysisl::Jobl](#)
- [Module:Bio::AnalysisParserl](#)
- [Module:Bio::AnalysisResultl](#)
- [Module:Bio::Annotatablel](#)
- [Module:Bio::Annotation](#)
- [Module:Bio::Annotation::AnnotationFactory](#)
- [Module:Bio::Annotation::Collection](#)
- [Module:Bio::Annotation::Comment](#)
- [Module:Bio::Annotation::DBLink](#)

M cont.

- [Module:Bio::Assembly::Singlet](#)
- [Module:Bio::Assembly::Tools::ContigSpectrum](#)
- [Module:Bio::Biblio](#)
- [Module:Bio::Biblio::Article](#)
- [Module:Bio::Biblio::BiblioBase](#)
- [Module:Bio::Biblio::Book](#)
- [Module:Bio::Biblio::BookArticle](#)
- [Module:Bio::Biblio::IO](#)
- [Module:Bio::Biblio::IO::medline2ref](#)
- [Module:Bio::Biblio::IO::medlinexml](#)
- [Module:Bio::Biblio::IO::pubmed2ref](#)
- [Module:Bio::Biblio::IO::pubmedxml](#)
- [Module:Bio::Biblio::Journal](#)
- [Module:Bio::Biblio::JournalArticle](#)
- [Module:Bio::Biblio::MedlineArticle](#)
- [Module:Bio::Biblio::MedlineBook](#)
- [Module:Bio::Biblio::MedlineBookArticle](#)
- [Module:Bio::Biblio::MedlineJournal](#)
- [Module:Bio::Biblio::MedlineJournalArticle](#)
- [Module:Bio::Biblio::Organisation](#)
- [Module:Bio::Biblio::Patent](#)
- [Module:Bio::Biblio::Person](#)
- [Module:Bio::Biblio::Proceeding](#)
- [Module:Bio::Biblio::Provider](#)
- [Module:Bio::Biblio::PubmedArticle](#)
- [Module:Bio::Biblio::PubmedBookArticle](#)
- [Module:Bio::Biblio::PubmedJournalArticle](#)
- [Module:Bio::Biblio::Ref](#)
- [Module:Bio::Biblio::Service](#)
- [Module:Bio::Biblio::TechReport](#)
- [Module:Bio::Biblio::Thesis](#)
- [Module:Bio::Biblio::WebResource](#)
- [Module:Bio::BioEntry](#)
- [Module:Bio::Cluster::ClusterFactory](#)
- [Module:Bio::Cluster::Familyl](#)
- [Module:Bio::Cluster::SequenceFamily](#)
- [Module:Bio::Cluster::UniGene](#)
- [Module:Bio::Cluster::UniGenel](#)
- [Module:Bio::Clusterl](#)
- [Module:Bio::ClusterlO](#)
- [Module:Bio::ClusterlO::dbsnp](#)
- [Module:Bio::ClusterlO::unigene](#)
- [Module:Bio::CodonUsage::IO](#)
- [Module:Bio::CodonUsage::Table](#)
- [Module:Bio::Coordinate](#)
- [Module:Bio::Coordinate::Chain](#)
- [Module:Bio::Coordinate::Collection](#)
- [Module:Bio::Coordinate::ExtrapolatingPair](#)
- [Module:Bio::Coordinate::GeneMapper](#)
- [Module:Bio::Coordinate::Graph](#)
- [Module:Bio::Coordinate::Mapperl](#)

M cont.

- [Module:Bio::DB::BioDB](#)
- [Module:Bio::DB::BioFetch](#)
- [Module:Bio::DB::BioSeqDatabase](#)
- [Module:Bio::DB::BioSQL::AnnotationCollectionAdaptor](#)
- [Module:Bio::DB::BioSQL::BaseDriver](#)
- [Module:Bio::DB::BioSQL::BasePersistenceAdaptor](#)
- [Module:Bio::DB::BioSQL::BioNamespaceAdaptor](#)
- [Module:Bio::DB::BioSQL::BiosequenceAdaptor](#)
- [Module:Bio::DB::BioSQL::ClusterAdaptor](#)
- [Module:Bio::DB::BioSQL::CommentAdaptor](#)
- [Module:Bio::DB::BioSQL::DBAdaptor](#)
- [Module:Bio::DB::BioSQL::DBLinkAdaptor](#)
- [Module:Bio::DB::BioSQL::LocationAdaptor](#)
- [Module:Bio::DB::BioSQL::MultiDB](#)
- [Module:Bio::DB::BioSQL::mysql::AnnotationCollectionAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::mysql::BasePersistenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::mysql::BiosequenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::mysql::PathAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::mysql::SpeciesAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::mysql::TermAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::OntologyAdaptor](#)
- [Module:Bio::DB::BioSQL::Oracle::AnnotationCollectionAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Oracle::BasePersistenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Oracle::BiosequenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Oracle::PathAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Oracle::SpeciesAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Oracle::TermAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::PathAdaptor](#)
- [Module:Bio::DB::BioSQL::Pg::AnnotationCollectionAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Pg::BasePersistenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Pg::BiosequenceAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Pg::PathAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Pg::SpeciesAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::Pg::TermAdaptorDriver](#)
- [Module:Bio::DB::BioSQL::PrimarySeqAdaptor](#)
- [Module:Bio::DB::BioSQL::ReferenceAdaptor](#)
- [Module:Bio::DB::BioSQL::RelationshipAdaptor](#)
- [Module:Bio::DB::BioSQL::SeqAdaptor](#)
- [Module:Bio::DB::BioSQL::SeqFeatureAdaptor](#)
- [Module:Bio::DB::BioSQL::SimpleValueAdaptor](#)
- [Module:Bio::DB::BioSQL::SpeciesAdaptor](#)
- [Module:Bio::DB::BioSQL::TermAdaptor](#)
- [Module:Bio::DB::CacheServer::SeqDB](#)
- [Module:Bio::DB::CUTG](#)
- [Module:Bio::DB::DBAdaptorl](#)
- [Module:Bio::DB::DBContextl](#)
- [Module:Bio::DB::DBD](#)
- [Module:Bio::DB::DBFetch](#)
- [Module:Bio::DB::DBl](#)
- [Module:Bio::DB::DBl::base](#)
- [Module:Bio::DB::DBl::mysql](#)

CPAN

Comprehensive Perl Archive Network

Instalação do BioPerl utilizando o módulo **CPAN.pm**

Instalação:

```
Darwin:~ Tatiana$ cpan[4]> q

Darwin:~ Tatiana$ sudo perl -MCPAN -e shell

cpan[1]> o conf prerequisites_policy follow
      prerequisites_policy [follow]
Please use 'o conf commit' to make the config permanent!

cpan[2]> o conf commit
commit: wrote '/Users/Tatiana/.cpan/CPAN/MyConfig.pm'

cpan[3]> d /bioperl/
```

CPAN

Comprehensive Perl Archive Network

Instalação do BioPerl utilizando o módulo [CPAN.pm](#)

Continuar instalação:

```
Writing /Users/tatiana/.cpan/Metadata
Distribution      BOZO/Fry-Lib-BioPerl-0.15.tar.gz
Distribution      CDRAUG/BioPerl-1.7.4.tar.gz
Distribution      CDRAUG/Dist-Zilla-PluginBundle-BioPerl-0.27.tar.gz
Distribution      CJFIELDS/BioPerl-1.007002.tar.gz
Distribution      CJFIELDS/BioPerl-1.6.924.tar.gz
Distribution      CJFIELDS/BioPerl-1.7.7.tar.gz
Distribution      CJFIELDS/BioPerl-1.7.8.tar.gz
Distribution      CJFIELDS/BioPerl-DB-1.006900.tar.gz
Distribution      CJFIELDS/BioPerl-Network-1.006902.tar.gz
Distribution      CJFIELDS/BioPerl-Run-1.007002.tar.gz
Distribution      CJFIELDS/BioPerl-Run-1.007003.tar.gz
Distribution      CJFIELDS/Bundle-BioPerl-2.1.9.tar.gz
12 items found
```

CPAN

Comprehensive Perl Archive Network

Instalação do BioPerl utilizando o módulo [CPAN.pm](#)

Continuar instalação:

```
Distribution    CJFIELDS/BioPerl-1.007002.tar.gz
Distribution    CJFIELDS/BioPerl-1.6.924.tar.gz
Distribution    CJFIELDS/BioPerl-1.7.7.tar.gz
Distribution    CJFIELDS/BioPerl-1.7.8.tar.gz
Distribution    CJFIELDS/BioPerl-DB-1.006900.tar.gz
Distribution    CJFIELDS/BioPerl-Network-1.006902.tar.gz
Distribution    CJFIELDS/BioPerl-Run-1.007002.tar.gz
Distribution    CJFIELDS/BioPerl-Run-1.007003.tar.gz
Distribution    CJFIELDS/Bundle-BioPerl-2.1.9.tar.gz
12 items found
```

```
cpan[4]> force install CJFIELDS/BioPerl-1.7.8.tar.gz
```

Instalação do BioPerl utilizando o módulo **CPAN.pm**

Depois de ~10 min, ~18 min, ~25 min, haverá necessidade de interação.

Se tudo der certo (~30 min):

```
Distribution      CJFIELDS/BioPerl-1.007002.tar.gz
Distribution      CJFIELDS/BioPerl-1.6.924.tar.gz
Distribution      CJFIELDS/BioPerl-1.7.7.tar.gz
Distribution      CJFIELDS/BioPerl-1.7.8.tar.gz
Distribution      CJFIELDS/BioPerl-DB-1.006900.tar.gz
Distribution      CJFIELDS/BioPerl-Network-1.006902.tar.gz
Distribution      CJFIELDS/BioPerl-Run-1.007002.tar.gz
Distribution      CJFIELDS/BioPerl-Run-1.007003.tar.gz
Distribution      CJFIELDS/Bundle-BioPerl-2.1.9.tar.gz
12 items found
```

```
cpan[4]> force install CJFIELDS/BioPerl-1.7.8.tar.
```

Muitas linhas

```
cpan[5]> q
```


CPAN vs CPANM

CPAN

`cpan` :

- Produz saídas detalhadas, mostrando cada etapa:
 - Download
 - Verificação de checksum
 - Execução do instalador
 - Parsing
 - Carregamento de dependências
- Pode gerar **centenas de linhas de saída**, especialmente se houver muitas dependências.

CPAN vs CPANM

CPAN minus

`cpanm` :

- Fornece um resumo mais conciso das etapas gerais:
 - `Fetching`
 - `Configuring`
 - `Building and testing`
 - `Done`
- Ideal para quem prefere saídas limpas e informações apenas essenciais.
- Não está instalado em `core`

CPANM

Instalação do CPANM utilizando o módulo CPAN

Instalação:

```
Darwin:~ Tatiana$ sudo perl -MCPAN -e shell
```

```
cpan[1]> install App::cpanminus
```

```
cpan[2]> q
```

```
Darwin:~ Tatiana$ cpanm --version
```

BIOPERL

Instalação do BIOPERL utilizando o módulo CPANM

Instalação:

```
Darwin:~ Tatiana$ cpanm Bio::Perl
```

BIOPERL

Instalação do BIOPERL utilizando o módulo CPANM

Instalação:

```
Darwin:~ Tatiana$ cpanm Bio::Perl
...
Building and testing Bio-Procedural-1.7.4 ... FAIL
! Installing Bio::Perl failed. See
/Users/tatiana/.cpanm/work/1731987489.73333/build.log for
details. Retry with --force to force install it.
```

BIOPERL

Instalação do BIOPERL utilizando o módulo CPANM

Instalação:

```
Darwin:~ Tatiana$ cpanm Bio::Perl
...
Building and testing Bio-Procedural-1.7.4 ... FAIL
! Installing Bio::Perl failed. See
/Users/tatiana/.cpanm/work/1731987489.73333/build.log for
details. Retry with --force to force install it.
Darwin:~ Tatiana$
Darwin:~ Tatiana$ sudo cpanm Bio::Perl
```

BIOPERL

Instalação interativa

No terminal, verificar a instalação do BioPerl

```
Darwin:~ Tatiana$ perl -mBio::Seq -e 'print "Modulo  
presente\n"'
```

-e script uma linha de script

-[mM][-]modulo executar "usar/não usar módulo..." antes de executar script

