

Introdução à Programação de Computadores para Biologia

BIOPERL

Aula 15

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

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

BIOPERL

Caso não esteja instalado

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

Instalação ():

```
Darwin:~ Tatiana$ sudo perl -MCPAN -e shell  
cpan[1]> force install Bio::Perl
```

BIOPERL

Programação Orientada a Objetos

- Declarações de variáveis
- Corpo do programa
- Subrotinas

Programação Estruturada

BIOPERL

Programação Orientada a Objetos

- Declarações de variáveis
- Corpo do programa
- Subrotinas

Programação Estruturada

- Corpo do programa

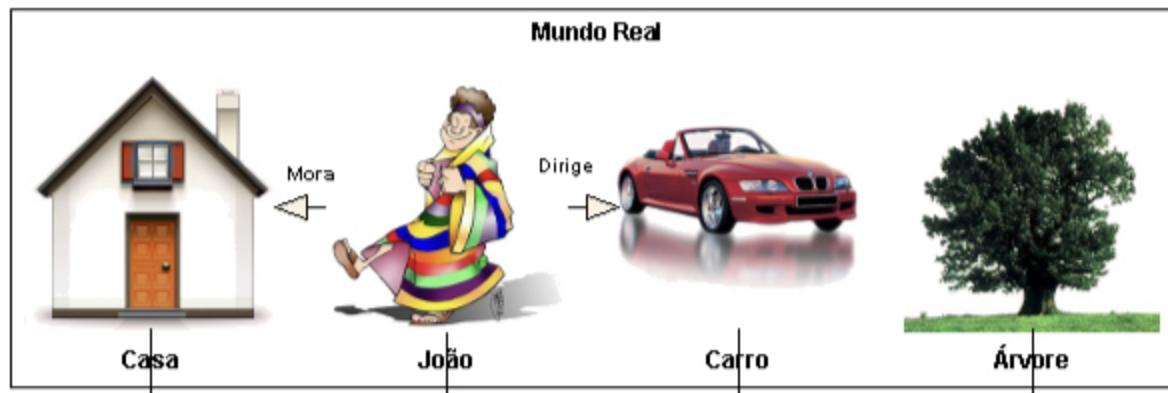
- Classe 1
 - variáveis
 - métodos

- Classe 2
 - variáveis
 - métodos

Programação Orientada a Objetos

BIOPERL

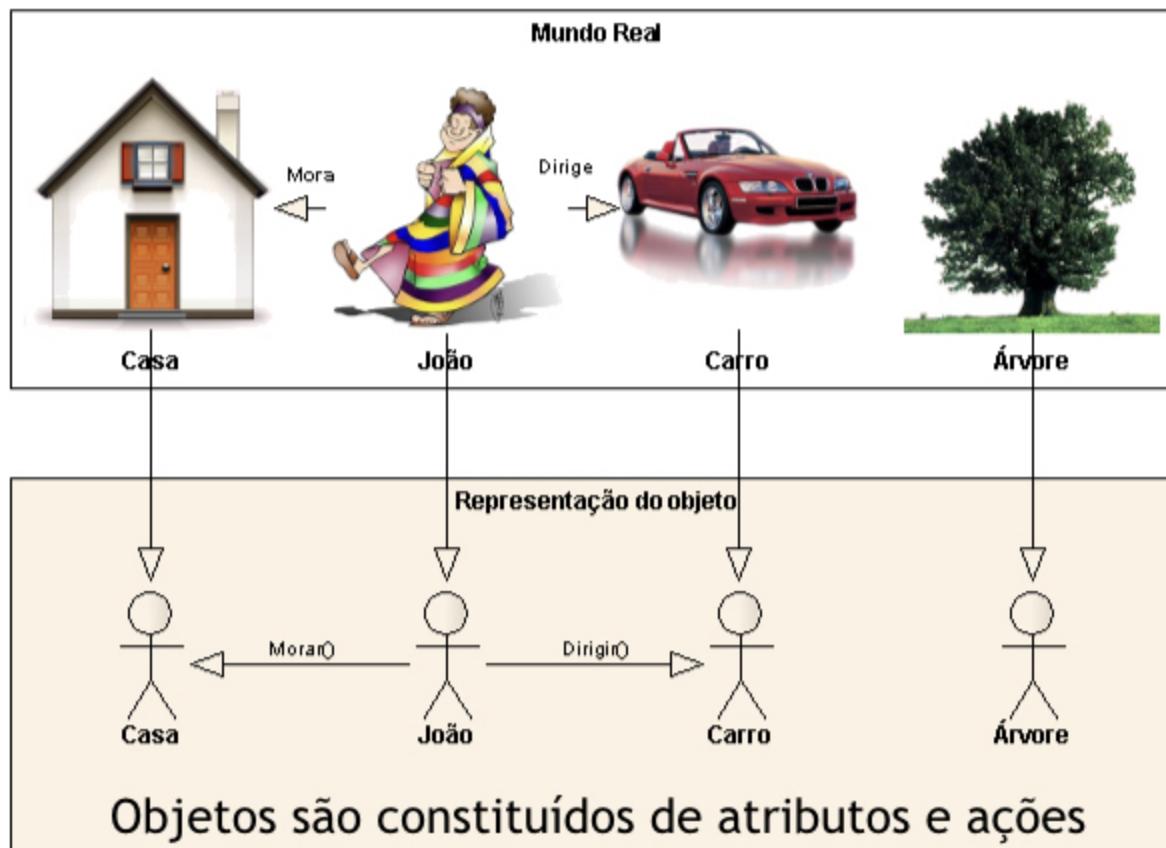
Programação Orientada a Objetos



Objetos são constituídos de atributos e ações

BIOPERL

Programação Orientada a Objetos



BIOPERL

Programação Orientada a Objetos



Objeto FLOR

ATRIBUTOS:

- cor
- forma
- número de pétalas
- tamanho

MÉTODOS:

- abrir()
- murchar()

BIOPERL

Programação Orientada a Objetos



BIOPERL

Programação Orientada a Objetos

CLASSE FLORES



CLASSE ARVORES

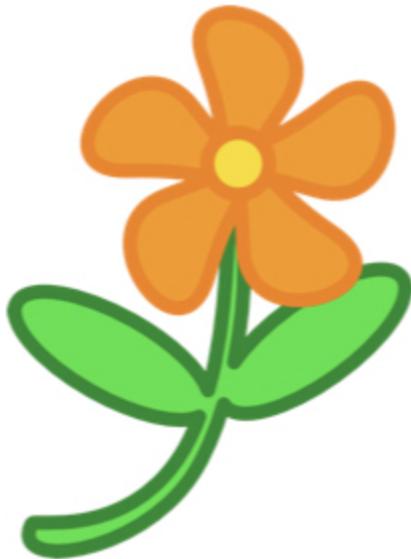


CLASSE GRAMÍNEAS



BIOPERL

Programação Orientada a Objetos



CLASSE FLORES

ATRIBUTOS OBJETO FLOR:

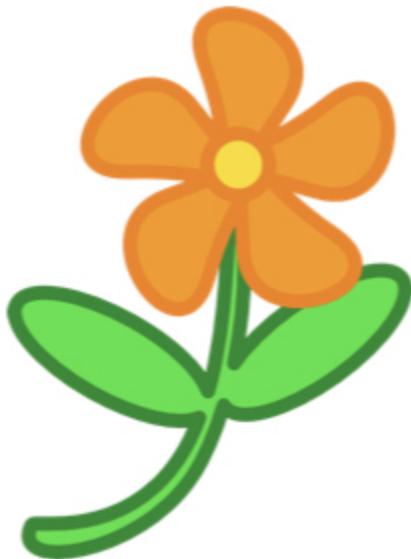
- cor
- forma
- número de pétalas
- tamanho

MÉTODOS OBJETO FLOR:

- abrir()
- murchar()

BIOPERL

Programação Orientada a Objetos



BIOPERL

Programação Orientada a Objetos

ATGCGTCGAGAT
GTCTACGTAGTG
CATGCTCGTAGC
TAGCTAGCTAGC
TAGCTAGCTACG
TAGCTACGTAGC
TACTATATATAG
CTATATATGAGC
GATACGGATATA
TTAAAAAATTGGC
CCTATCCCTATA

DNA

ATRIBUTOS OBJETO SEQUENCIA:

- sequencia
- ID
- descrição
- acesso

MÉTODOS OBJETO SEQUENCIA:

- criar()
- imprimir()
- calcular tamanho()

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

```
#!/usr/bin/perl/  
  
## Introducao ao uso do modulo Bio::Seq  
## Este modulo (Bio/Seq.pm) cria o objeto sequencia  
  
use Bio::Seq;  
  
my $seq_obj = Bio::Seq->new(-seq => "aaaatggggggggggggccccc",  
                                -alphabet => "dna",  
                                -desc => "Exemplo",  
                                -display_id => "01",  
                                -accession_number => "GB01");  
exit;
```

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [bioseq.pl](#)
4. Copiar **exemplo01** da página da disciplina.
5. Executar o script e determinar sua função.

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

```
#!/usr/bin/perl/  
  
use strict;  
use Bio::Seq;  
  
# criar o objeto sequencia  
my $seq_obj = Bio::Seq->new(  
    -seq => "aaaatggggggggggcccgtt",  
    -alphabet => "dna",  
    -desc => "Exemplo",  
    -display_id => "01",  
    -accession_number => "GB01");  
  
# apos criar o objeto, podemos usar varios metodos  
# exemplo, imprimir sequencia em formato fasta  
  
print ">" . $seq_obj->accession_number() .  
      " " . $seq_obj->desc() . "\n";  
  
print $seq_obj->seq . "\n";  
  
exit;
```

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [bioseq.pl](#)
4. Copiar **exemplo01** da página da disciplina.
5. Executar o script e determinar sua função.
6. Modificar o script para imprimir 10 nucleotídeos por linha.

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

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


use strict;
use Bio::Seq;

# criar o objeto sequencia
my $seq_obj = Bio::Seq->new(-seq => "aaaatggggggggggcccgatctgcgtcg",
                             -alphabet => "dna",
                             -desc => "Exemplo",
                             -display_id => "01",
                             -accession_number => "GB01");

# apos criar o objeto, podemos usar varios metodos
# exemplo, imprimir sequencia em formato fasta

print ">" . $seq_obj->accession_number() . " " . $seq_obj->desc() . "

for (my $i = 0; $i < $seq_obj->length(); $i += 10) {
    my $end = $i + 10;
    print $seq_obj->subseq($i, $end) . "\n";
}

exit;
```

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

```
Darwin:~ Tatiana$ perl bioseq.pl  
>GB01 Exemplo
```

```
----- EXCEPTION: Bio::Root::Exception -----  
MSG: Bad end parameter (51). End must be less than the total length of the sequence  
STACK: Error::throw  
STACK: Bio::Root::Root::throw /Library/Perl/5.18/Bio/Root/Root.pm:447  
STACK: Bio::PrimarySeq::subseq /Library/Perl/5.18/Bio/PrimarySeq.pm:447  
STACK: Bio::Seq::subseq /Library/Perl/5.18/Bio/Seq.pm:630  
STACK: sub.pl:23
```

BIOPERL Bio::Seq

```
#!/usr/bin/perl

use strict;
use Bio::Seq;

# criar o objeto sequencia
my $seq_obj = Bio::Seq->new(-seq => "aaaatggggggggggcccgatctgcgtcg"
                            -alphabet => "dna",
                            -desc => "Exemplo",
                            -display_id => "01",
                            -accession_number => "GB01");

# apos criar o objeto, podemos usar varios metodos
# exemplo, imprimir sequencia em formato fasta

print ">" . $seq_obj->accession_number() . " " . $seq_obj->desc() . ""

for (my $i = 0; $i < $seq_obj->length(); $i += 10) {
    my $end = $i + 10;
    if ($end > $seq_obj->length()) {
        $end = $seq_obj->length();
    }
    print $seq_obj->subseq($i, $end) . "\n";
}

exit;
```

BIOPERL Bio::Seq

```
#!/usr/bin/perl

use strict;
use Bio::Seq;

# criar o objeto sequencia
my $seq_obj = Bio::Seq->new(-seq => "aaaatggggggggggcccgatctgcgtcg"
                           -alphabet => "dna",
                           -desc => "Exemplo",
                           -display_id => "01",
                           -accession_number => "GB01");

# apos criar o objeto, podemos usar varios metodos
# exemplo, imprimir sequencia em formato fasta

print ">" . $seq_obj->accession_number() . " " . $seq_obj->desc() . ""

for (my $i = 1; $i < $seq_obj->length(); $i += 10) {
    my $end = $i + 10;
    if ($end > $seq_obj->length()) {
        $end = $seq_obj->length();
    }
    print $seq_obj->subseq($i, $end) . "\n";
}

exit;
```

BIOPERL Bio::Seq

Atributos e métodos para objeto sequência

Name	Returns	Example	Note
accession_number	identifier	\$acc = \$so->accession_number	get or set an identifier
alphabet	alphabet	\$so->alphabet('dna')	get or set the alphabet ('dna','rna','protein')
authority	authority, if available	\$so->authority("FlyBase")	get or set the organization
desc	description	\$so->desc("Example 1")	get or set a description
display_id	identifier	\$so->display_id("NP_123456")	get or set an identifier
division	division, if available (e.g. PRI)	\$div = \$so->division	get division (e.g. "PRI")
get_dates	array of dates, if available	@dates = \$so->get_dates	get dates
get_secondary_acccessions	array of secondary accessions, if available	@accs = \$so->get_secondary_acccessions	get other identifiers
is_circular	Boolean	if \$so->is_circular (#)	get or set
keywords	keywords, if available	@array = \$so->keywords	get or set keywords
length	length, a number	\$len = \$so->length	get the length
molecule	molecule type, if available	\$type = \$so->molecule	get molecule (e.g. "RNA", "DNA")
namespace	namespace, if available	\$so->namespace("Private")	get or set the name space
new	Sequence object	\$so = Bio::Seq->new(-seq => "MPQRAS")	create a new one, see Bio::Seq for more
pid	pid, if available	\$pid = \$so->pid	get pid
primary_id	identifier	\$so->primary_id(12345)	get or set an identifier
revcom	Sequence object	\$so2 = \$so1->revcom	Reverse complement
seq	sequence string	\$seq = \$so->seq	get or set the sequence
seq_version	version, if available	\$so->seq_version("1")	get or set a version
species	Species object	\$species_obj = \$so->species	See Bio::Species for more
subseq	sequence string	\$string = \$seq_obj->subseq(10,40)	Arguments are start and end
translate	protein Sequence object	\$prot_obj = \$dnaseq_obj->translate	
trunc	Sequence object	\$so2 = \$so1->trunc(10,40)	Arguments are start and end

BIOPERL Bio::Seq

Input e Output de Sequências

```
#!/usr/bin/perl/  
  
use strict;  
use Bio::SeqIO;  
  
### Argumento  
my $fastafilename = $ARGV[0];  
my $outfile = $ARGV[1];  
  
### Criar objeto ENTRADA  
my $all_seqs_obj = Bio::SeqIO->new(-file => "<$fastafilename",  
                                     -format => 'Fasta');  
### Criar objeto SAIDA  
my $out_obj = Bio::SeqIO->new(-file => ">$outfile",  
                               -format => 'Fasta');  
  
### Acessar todos os elementos da entrada  
### metodo next_seq, a cada iteracao do while pega o proximo objeto s  
while (my $seq_obj = $all_seqs_obj->next_seq()) {  
    $out_obj->write_seq($seq_obj);  
}  
exit;
```

BIOPERL Bio::Seq

Input e Output de Sequências

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [genbank.pl](#)
4. Copiar **exemplo03** da página da disciplina.
5. Executar o script com o arquivo Genbank_flatfile.txt como input e dar um novo nome para o output:

```
perl genbank.pl Genbank_flatfile.txt teste_output.txt
```

BIOPERL Bio::Seq

Input e Output de Sequências

```
#!/usr/bin/perl/  
  
use strict;  
use Bio::SeqIO;  
  
### Argumento  
my $infile = $ARGV[0];  
my $outfile = $ARGV[1];  
  
### Criar objeto ENTRADA  
my $all_seqs_obj = Bio::SeqIO->new(-file => "<$infile",  
                                     -format => 'Genbank');  
### Criar objeto SAIDA  
my $out_obj = Bio::SeqIO->new(-file => ">$outfile",  
                               -format => 'Fasta');  
  
### Acessar todos os elementos da entrada  
### metodo next_seq, a cada iteracao do while pega o proximo objeto s  
  
while (my $seq_obj = $all_seqs_obj->next_seq()) {  
    $out_obj->write_seq($seq_obj);  
}  
exit;
```

BIOPERL Bio::Seq

Input e Output de Sequências

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [getSeqs.pl](#)
4. Copiar **exemplo03** da página da disciplina.
5. Modificar o **exemplo03** para usar como entrada o arquivo dmel-gene-r5.45.fasta (mudar o atributo -format no script)

BIOPERL Bio::Seq Input e Output

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

use strict;
use Bio::SeqIO;

### Argumento
my $infile = $ARGV[0];
my $outfile = $ARGV[1];

### Criar objeto ENTRADA
my $all_seqs_obj = Bio::SeqIO->new(-file => "<$infile",
                                     -format => 'Fasta');

### Criar objeto SAIDA
my $out_obj = Bio::SeqIO->new(-file => ">$outfile",
                               -format => 'Fasta');

### Acessar todos os elementos da entrada
### metodo next_seq, a cada iteracao do while pega o proximo objeto s

while (my $seq_obj = $all_seqs_obj->next_seq()) {
    $out_obj->write_seq($seq_obj);
}

exit;
```

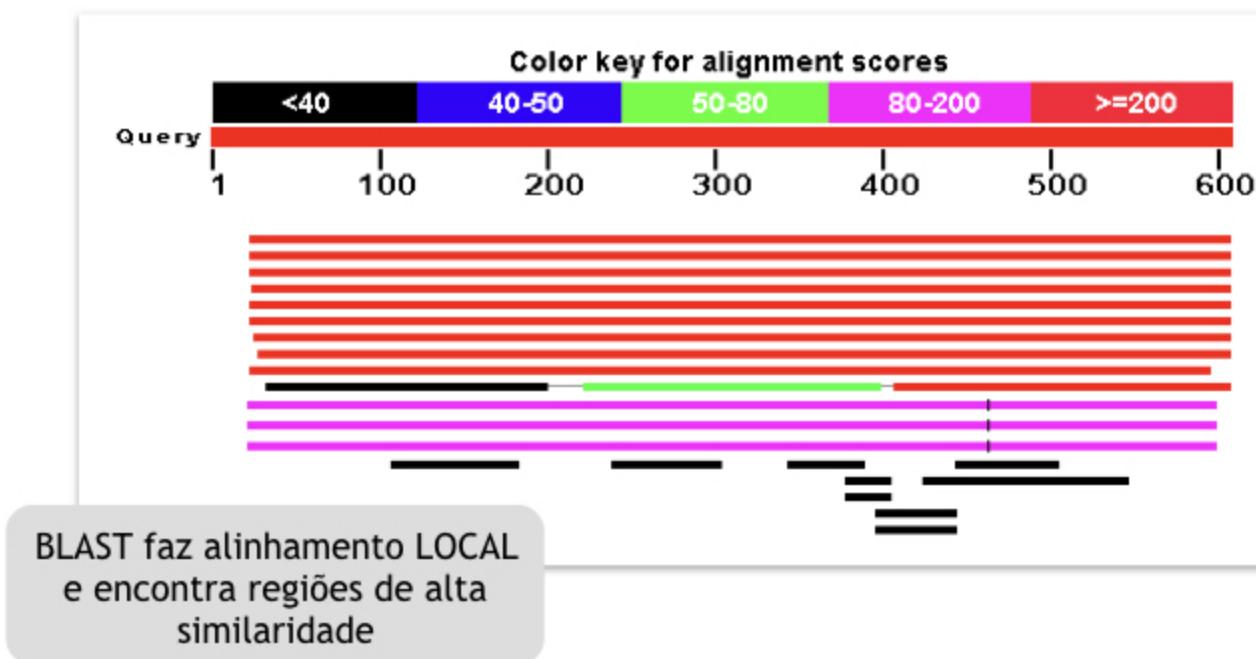
BIOPERL Bio::SeqIO Input e Output

Modificar [getSeqs.pl](#) para imprimir o complemento reverso das sequências.

```
#!/usr/bin/perl/  
  
use strict;  
use Bio::SeqIO;  
  
### Argumento  
my $fastafile = $ARGV[0];  
my $outfile = $ARGV[1];  
  
### Criar objeto ENTRADA  
my $all_seqs_obj = Bio::SeqIO->new(-file => "<$fastafile",  
                                     -format => 'Genbank');  
### Criar objeto SAIDA  
my $out_obj = Bio::SeqIO->new(-file => ">$outfile",  
                               -format => 'Fasta');  
  
### Acessar todos os elementos da entrada  
### metodo next_seq, a cada iteracao do while pega o proximo objeto s  
  
while (my $seq_obj = $all_seqs_obj->next_seq()) {  
    $out_obj->write_seq($seq_obj->revcom());;  
}  
  
exit;
```

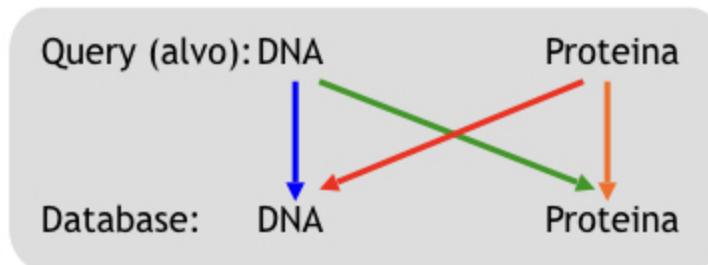
BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool



BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool



blastn - nucleotídeos vs. nucleotídeos

blastp - proteína vs. proteína

blastx - query traduzida vs. banco de proteínas

tblastn - proteína vs. banco de nucleotídeos traduzidos

tblastx - query traduzida vs. banco de nucleotídeos traduzidos

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

The screenshot shows the NCBI BLAST search results for the query sequence ref|YP_094059|. The results are displayed in two main sections: Text and Graphic Summary.

Text Section:

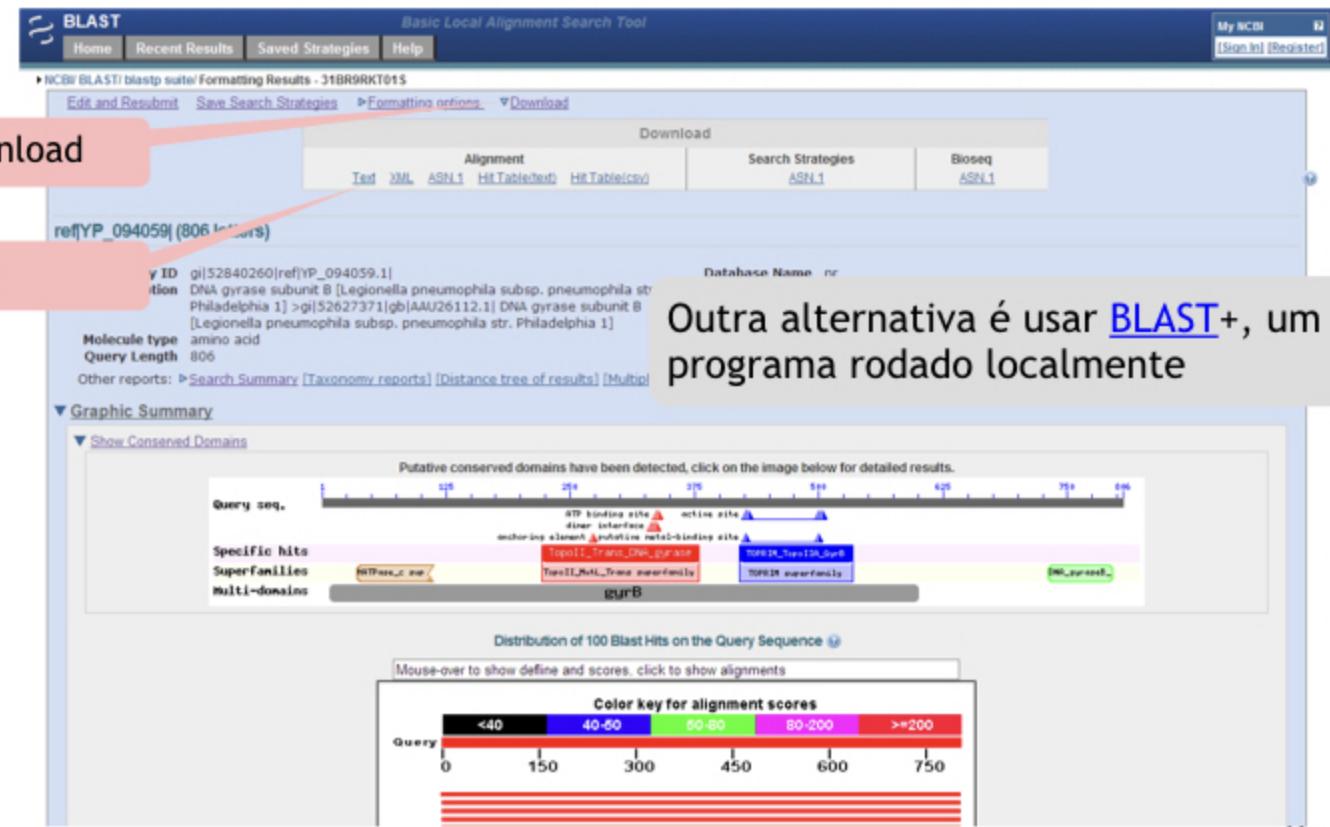
- Download:** Options for Text, XML, ASN.1, HitTable(tab), HitTable(csv), Alignment, Search Strategies (ASN.1), and Bioseq (ASN.1).
- ref|YP_094059| (806 letters)**
- Query ID:** gi|52840260|ref|YP_094059.1|
- Description:** DNA gyrase subunit B [Legionella pneumophila subsp. pneumophila str. Philadelphia 1] >gi|52627371|gb|AAU26112.1| DNA gyrase subunit B [Legionella pneumophila subsp. pneumophila str. Philadelphia 1]
- Molecule type:** amino acid
- Query Length:** 806
- Other reports:** Search Summary, Taxonomy reports, Distance tree of results, Multiple alignment.

Graphic Summary Section:

- Show Conserved Domains:** Putative conserved domains have been detected. Click on the image below for detailed results.
- Query seq.:** A sequence diagram showing the query sequence with various domain predictions. Annotations include: ATP binding site, dimer interface, active site, anchoring element, divalent metal-binding site, TopoII_Trans_DNA_buried, TopoII_Mot1_Trans_overfamily, TopoII_overfamily, and gyrB.
- Distribution of 100 Blast Hits on the Query Sequence:** A chart showing the distribution of 100 blast hits along the query sequence. The x-axis represents the position from 0 to 750. The color key for alignment scores is: <40 (black), 40-60 (blue), 60-80 (green), 80-200 (yellow), and >=200 (red). The chart shows several red bars indicating high-scoring alignments.

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool



BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

Query= chom_c1
(868 letters)

Query

Database: insect-all-transcript.fasta
277,359 sequences; 399,176,351 total letters

Searching.....done

Resultados

Sequences producing significant alignments:

Score E
(bits) Value N

FBtr0283794 type=mRNA; loc=2:complement(11218186..11218630,1...	450	e-125	2
FBtr0190005 type=mRNA; loc=scaffold_0:join(9251337..9251506,...	450	e-125	2
FBtr0133034 type=mRNA; loc=scaffold_4770:join(1753590..17537...	448	e-125	2
FBtr0256670 type=mRNA; loc=3R:join(17445059..17445228,174452...	448	e-125	2
FBtr0253055 type=mRNA; loc=scf2_1100000004921:complement(357...	443	e-123	2
FBtr0122278 type=mRNA; loc=scaffold_13340:join(1421004..1421...	442	e-123	2
FBtr0078712 type=mRNA; loc=3R:join(1480867..1481097,1481156....	442	e-123	2
FBtr0219717 type=mRNA; loc=3R:join(1515828..1515997,1516056....	441	e-123	2
FBtr0193814 type=mRNA; loc=scaffold_6:join(1595827..1595996,...	441	e-123	2
FBtr0153790 type=mRNA; loc=scaffold_14906:complement(6541574...	440	e-122	2

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
>FBtr0283794 type=mRNA; species=Dpse;  
...
```

Descrição do best hit

```
Score = 450 bits (977), Expect = e-125  
Identities = 188/267 (70%), Positives = 211/267 (79%)  
Frame = -3 / +1
```

Dados do HSP 1

```
Query: 866 GTGVEQSEESRNKFLVEFQENYGNINQQLREVESLINGLQVPMSPYYLGNTTYLAQEITQ 687  
       GT E EESRNKF+ +FQE +G IN QLREVE LINGLQVP + Y LGNT YLAQE +Q  
Sbjct: 82  GTDGEVVEESRNKFVHDFQEKGAINSQLREVEQLINGLQVPPPTAYSLGNTAYLAQETSQ 261
```

...

Alinhamento do HSP 1

```
Score = 188 bits (404), Expect(2) = 2e-62  
Identities = 93/152 (61%), Positives = 106/152 (69%)  
Frame = +1 / -2
```

Dados do HSP 2

```
Query: 70  QFLHSS*KGSKVLKSRQVGGKPFTETYLPHLLQGSXXXXXXXXSQVI*LFTANSGSVGEKK 249  
       Q HSS KGS+V +SR VGG+P YL LQGS+ SQV *LFTA SGSGVGEKK  
Sbjct: 878 QLRHSSWKGSRVRRSRHVGGRPLAMYLLQLQGSRKSLMWLFQVT*LFTARSGSVGEKK 699
```

Alinhamento do HSP 2

```
>FBtr0190005 type=mRNA; species=Dper;
```

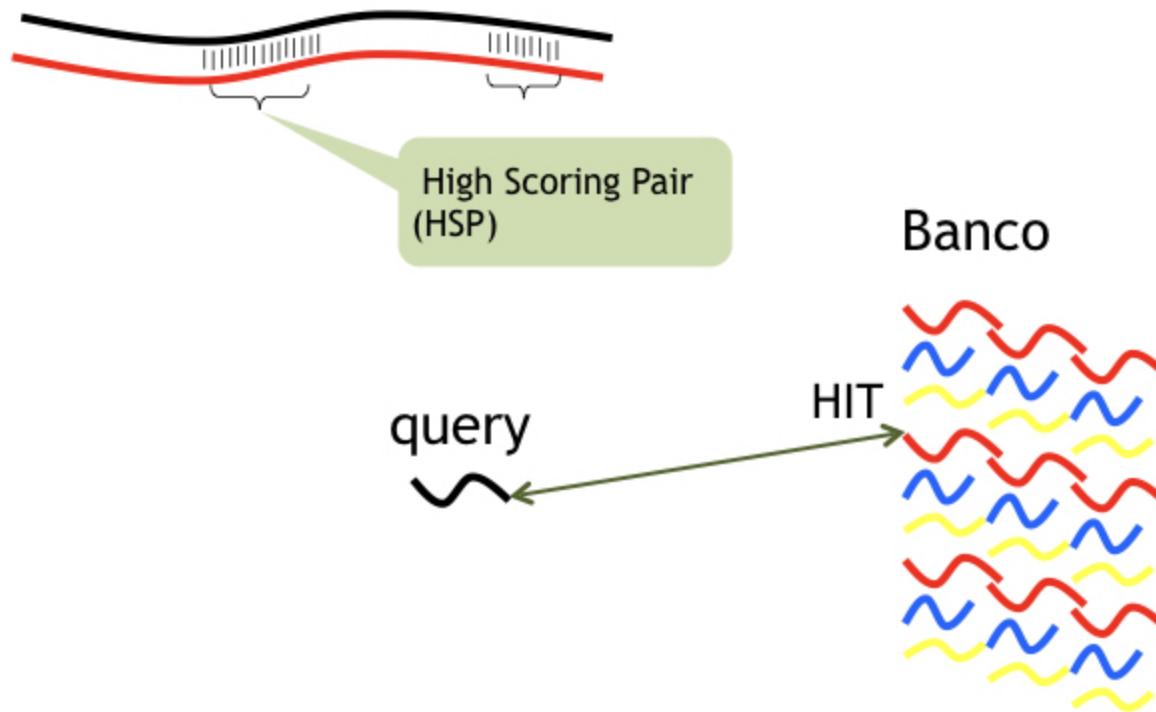
Descrição do segundo hit

```
Score = 450 bits (977), Expect = e-125  
Identities = 188/267 (70%), Positives = 211/267 (79%)  
Frame = -3 / +1
```

Dados do HSP 1

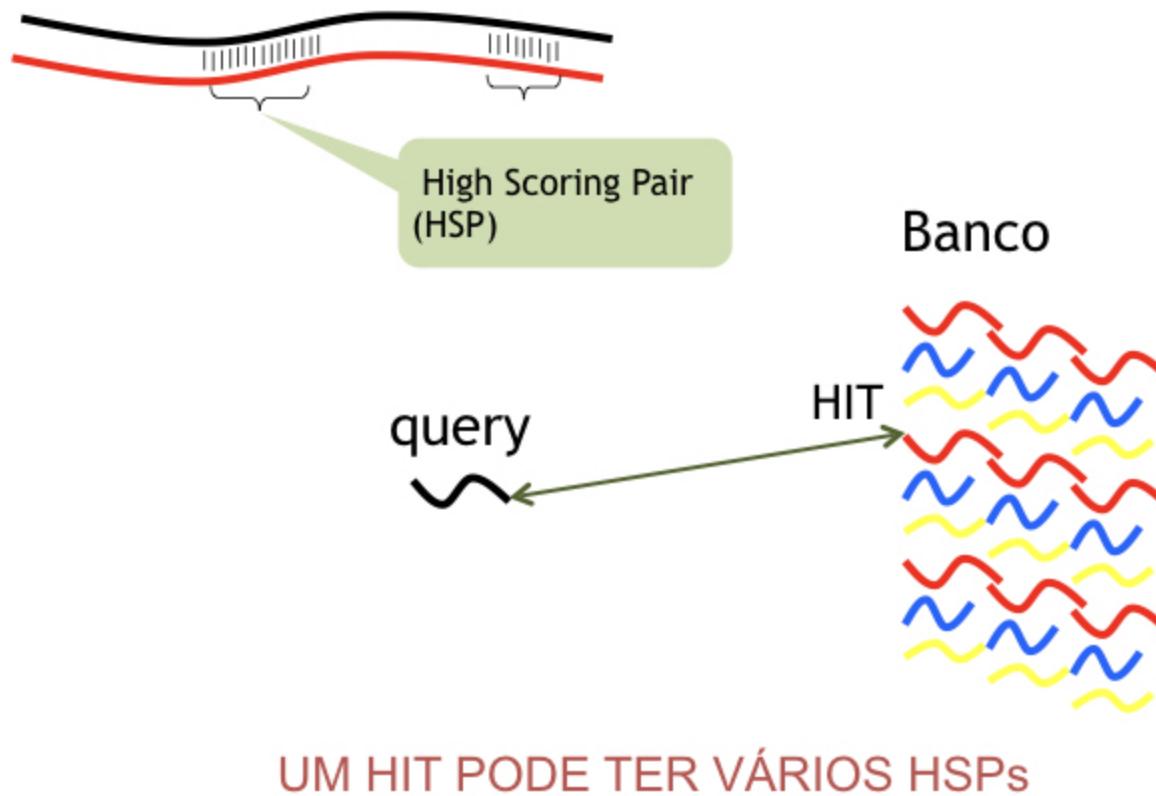
BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool



BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool



BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

1. Baixar o arquivo BLASToutput.txt, na página da disciplina e abrir com um editor de texto simples.
2. Localizar o resultado, os hits para cada resultado e as HSPs para cada hit.

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
BLASTN 2.2.28+
...
Query= c262873_g1_i2
Length=878
RID: 765JCZBV014

Sequences producing significant alignments:
Score      E
(Bits)    Value

emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold... 1299  0.0
...
gb|JQ259057.1| Aphis glycines 28S ribosomal RNA gene, partial se... 754   0.0

>emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold00093 complete
sequence Length=30980

Score = 1299 bits (703),  Expect = 0.0 Identities = 821/879 (93%), Gaps = 3/879 (0%) Strand=Plus/Minus

Query 1      CTGGGCAGAAATCACATTGCGTCAACACCCTTTCTGGCCATCGCAATGCTATGTTTAA 60
||||||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
Sbjct 29578  CTGGGCAGAAATCACATTGCGTCAACACCCTTTCTGGCCATCGCAATGCTATGTTTAA 29519
...

Score = 1005 bits (544),  Expect = 0.0 Identities = 767/877 (87%), Gaps = 5/877 (1%) Strand=Plus/Minus

Query 4      GGCAGAAAATCACATTGCGTCAACACCCTTTCTGGCCATCGCAATGCTATGTTTAA 63
||||||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
Sbjct 17861  GGCAGAAAATCACATTGCGTTAACACTACTTTCTGGCTATCGCAATGCTATGTTTAA 17802
...

>gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 1919WRS large subunit
ribosomal RNA gene, partial sequence Length=3291

Score = 1299 bits (703),  Expect = 0.0 Identities = 822/880 (93%), Gaps = 5/880 (1%) Strand=Plus/Minus

Query 1      CTGGGCAGAAATCACATTGCGTCAACACCCTTTCTGGCCATCGCAATGCTATGTTTAA 60
||||||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
Sbjct 2147   CTGGGCAGAAATCACATTGCGTCAACACCCTTTCTGGCCATCGCAATGCTATGTTTAA 2088
```

BIOPERL eBLAST

BLAST - Basic Local Alignment and Search Tool

BLASTN 2.2.28+
...
Query= c262873_g1_i2
Length=878
RID: 765JCZBV014

Sequences producing significant alignments:	Score (Bits)	E Value
emb FO905808.1 Leptosphaeria biglobosa Thlaspii ibcn65_scaffold...	1299	0.0
gb JQ259057.1 Aphis glycines 28S ribosomal RNA gene, partial se...	754	0.0

>emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold00093 complete sequence Length=30980

next_hit()

Score = 1299 bits (703), Expect = 0.0 Identities = 821/879 (93%), Gaps = 3/879 (0%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 29578	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	29519

...

Score = 1005 bits (544), Expect = 0.0 Identities = 767/877 (87%), Gaps = 5/877 (1%) Strand=Plus/Minus

Query 4	GGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	63
Sbjct 17861	GGCAGAAATCACATTGCGTAAACACTACTTCTGGCTATCGCAATGCTATGTTTAA	17802

...

>gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 1919WRS large subunit ribosomal RNA gene, partial sequence Length=3291

Score = 1299 bits (703), Expect = 0.0 Identities = 822/880 (93%), Gaps = 5/880 (1%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 2147	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	2088

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

BLASTN 2.2.28+
...
Query= c262873_g1_i2
Length=878
RID: 765JCZBV014

Sequences producing significant alignments:	Score (Bits)	E Value
emb FO905808.1 Leptosphaeria biglobosa Thlaspii ibcn65_scaffold...	1299	0.0
gb JQ259057.1 Aphis glycines 28S ribosomal RNA gene, partial se...	754	0.0

>emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold00093 complete sequence Length=30980

next_hit()

Score = 1299 bits (703), Expect = 0.0 Identities = 821/879 (93%), Gaps = 3/879 (0%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 29578	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	29519

next_hsp()

Score = 1005 bits (544), Expect = 0.0 Identities = 767/877 (87%), Gaps = 5/877 (1%) Strand=Plus/Minus

Query 4	GGCAGAAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAATTAA	63
Sbjct 17861	GGCAGAAAATCACATTGCGTAAACACTACTTCTGGCTATCGCAATGCTATGTTTAATTAA	17802

...

>gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 1919WRS large subunit ribosomal RNA gene, partial sequence Length=3291

Score = 1299 bits (703), Expect = 0.0 Identities = 822/880 (93%), Gaps = 5/880 (1%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 2147	CTGGGCAGAAATCACATTGCGTCAACACCACCTTCTGGCCATCGCAATGCTATGTTTAA	2088

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

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

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
#!/usr/bin/perl/  
  
use strict;  
use Bio::Seq;  
use Bio::SeqIO;  
use Bio::SearchIO;  
  
## Declarando variaveis  
my $result_obj; my $hit; my $hsp;  
  
## Argumentos  
my $input = $ARGV[0];  
  
## Criando objeto blast  
my $report_obj = Bio::SearchIO->new(-format => 'blast',  
                                      -file    => $input);  
  
## Criando objeto result  
$result_obj = $report_obj->next_result;  
print "Query=", $result_obj->query_name, "\n";  
  
exit;
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
#!/usr/bin/perl

use Bio::SeqIO;
use Bio::SearchIO;

## Declarando variaveis
my $result_obj; my $hit; my $hsp;

## Argumentos
my $input = $ARGV[0];

## Criando objeto blast -> O OBJETO REPORT CONTEM TODAS AS QUERY'S
my $report_obj = Bio::SearchIO->new(-format => 'blast',
                                      -file   => $input);

## Criando objeto result -> O OBJETO RESULT CONTEM UMA QUERY E TODOS
$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

exit;
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
BLASTN 2.2.28+                                         next_result()

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

Database: Nucleotide collection (nt)
           29,249,348 sequences; 84,467,861,798 total letters

Query= c262873_g1_i2

Length=878

RID: 765JCZBV014                                         Score      E
                                                               (Bits)    Value

Sequences producing significant alignments:
emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold... 1299  0.0
gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 191... 1299  0.0
...

```

```
BLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

Database: Nucleotide collection (nt)
           29,249,348 sequences; 84,467,861,798 total letters

Query= c271591_g1_i2
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
BLASTN 2.2.28+
```

```
Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb  
Miller (2000), "A greedy algorithm for aligning DNA sequences", J  
Comput Biol 2000; 7(1-2):203-14.
```

```
Database: Nucleotide collection (nt)  
29,249,348 sequences; 84,467,861,798 total letters
```

```
Query= c262873_g1_i2
```

```
Length=878
```

```
RID: 765JCZBV014
```

	Score (Bits)	E Value
Sequences producing significant alignments:		
emb FO905808.1 Leptosphaeria biglobosa Thlaspii ibcn65_scaffold...	1299	0.0
gb EU223257.1 Phaeosphaeria avenaria f. sp. avenaria strain 191...	1299	0.0
...		

```
BLASTN 2.2.28+
```

```
next_result()
```

```
Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb  
Miller (2000), "A greedy algorithm for aligning DNA sequences", J  
Comput Biol 2000; 7(1-2):203-14.
```

```
Database: Nucleotide collection (nt)  
29,249,348 sequences; 84,467,861,798 total letters
```

```
Query= c271591_g1_i2
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
## Criando objeto result -> O OBJETO RESULT CONTEM UM QUERY E TODOS
## OS SEUS HITs e HSPs

$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

$hit = $result_obj->next_hit;
print "Hit=", $hit->name ,"\n";

$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

$hit = $result_obj->next_hit;
print "Hit=", $hit->name ,"\n";

exit;
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

`next_hit()`

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

```
$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

$hit = $result_obj->next_hit;
print "Hit=", $hit->name ,"\n";

$hsp = $hit->next_hsp;
print "Length=", $hsp->length('total'), "\t", "Percent_id=",
      $hsp->percent_identity, "\n";

$hsp = $hit->next_hsp;
print "Length=", $hsp->length('total'), "\t", "Percent_id=", $hsp->pe

$hit = $result_obj->next_hit;
print "Hit=", $hit->name ,"\n";

$result_obj = $report_obj->next_result;
print "Query=", $result_obj->query_name, "\n";

exit;
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

BLASTN 2.2.28+
...
Query= c262873_g1_i2
Length=878
RID: 765JCZBV014

Sequences producing significant alignments:	Score (Bits)	E Value
emb FO905808.1 Leptosphaeria biglobosa Thlaspii ibcn65_scaffold...	1299	0.0
gb JQ259057.1 Aphis glycines 28S ribosomal RNA gene, partial se...	754	0.0

>emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold00093 complete sequence Length=30980

Score = 1299 bits (703), Expect = 0.0 Identities = 821/879 (93%), Gaps = 3/879 (0%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCAC	TCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 29578	CTGGGCAGAAATCACATTGCGTCAACACCAC	TTCTGGCCATCGCAATGCTATGTTTAA	29519

Score = 1005 bits (544), Expect = 0.0 Identities = 767/877 (87%), Gaps = 5/877 (1%) Strand=Plus/Minus

Query 4	GGCAGAAATCACATTGCGTCAACACCAC	TCTGGCCATCGCAATGCTATGTTTAATT	63
Sbjct 17861	GGCAGAAATCACATTGCGTAAACACTACTT	CTGGCTATCGCAATGCTATGTTTAATT	17802

>gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 1919WRS large subunit ribosomal RNA gene, partial sequence Length=3291

Score = 1299 bits (703), Expect = 0.0 Identities = 822/880 (93%), Gaps = 5/880 (1%) Strand=Plus/Minus

Query 1	CTGGGCAGAAATCACATTGCGTCAACACCAC	TCTGGCCATCGCAATGCTATGTTTAA	60
Sbjct 2147	CTGGGCAGAAATCACATTGCGTCAACACCAC	TTCTGGCCATCGCAATGCTATGTTTAA	2088

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [blastParser.pl](#)
4. Copiar **exemplo05** da página da disciplina.
5. Rodar o script usando como entrada o arquivo BLASToutput.txt.
6. O que faz cada while no script?

```
#!/usr/bin/perl

use strict;
use Bio::Seq;
use Bio::SeqIO;
use Bio::SearchIO;

## Declarando variaveis
my $result_obj; my $hit; my $hsp;

## Argumentos
my $input = $ARGV[0];

## Criando objeto blast
my $report_obj = Bio::SearchIO->new(-format => 'blast',
                                      -file    => $input);

while( $result_obj = $report_obj->next_result ) {
    while( $hit = $result_obj->next_hit ) {
        while( $hsp = $hit->next_hsp ) {
            if ( $hsp->percent_identity > 60 ) {
                print "Query=", $result_obj->query_name,
                      "\t", "Hit=", $hit->name,
                      "\t", "Length=", $hsp->length('total'),
                      "\t", "Percent_id=", $hsp->percent_identity, "\n";
            }
        }
    }
}
exit;
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

Primeiro while:

```
BLASTN 2.2.28+                                         next_result()

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

Database: Nucleotide collection (nt)
           29,249,348 sequences; 84,467,861,798 total letters

Query= c262873_g1_i2

Length=878

RID: 765JCZBV014

Sequences producing significant alignments:          Score      E
                                                (Bits)  Value

emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold...  1299      0.0
gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 191...  1299      0.0
...
.

BLASTN 2.2.28+                                         next_result()

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

Database: Nucleotide collection (nt)
           29,249,348 sequences; 84,467,861,798 total letters

Query= c271591_g1_i2
```

BIOPERL e BLAST

BLAST - Basic Local Alignment and Search Tool

Primeiro while:

```
BLASTN 2.2.28+  
  
Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb  
Miller (2000), "A greedy algorithm for aligning DNA sequences", J  
Comput Biol 2000; 7(1-2):203-14.  
  
Database: Nucleotide collection (nt)  
          29,249,348 sequences; 84,467,861,798 total letters  
  
Query= c262873_g1_i2  
  
Length=878  
  
RID: 765JCZBV014  
  
Sequences producing significant alignments:  
Score      E  
(Bits)    Value  
  
emb|FO905808.1| Leptosphaeria biglobosa Thlaspii ibcn65_scaffold... 1299      0.0  
gb|EU223257.1| Phaeosphaeria avenaria f. sp. avenaria strain 191... 1299      0.0  
...  
...
```

```
BLASTN 2.2.28+
```

next_result()

```
Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb  
Miller (2000), "A greedy algorithm for aligning DNA sequences", J  
Comput Biol 2000; 7(1-2):203-14.
```

```
Database: Nucleotide collection (nt)  
          29,249,348 sequences; 84,467,861,798 total letters
```

```
Query= c271591_g1_i2
```

```
#!/usr/bin/perl

use strict;
use Bio::Seq;
use Bio::SeqIO;
use Bio::SearchIO;

## Declarando variaveis
my $result_obj; my $hit; my $hsp;

## Argumentos
my $input = $ARGV[0];

## Criando objeto blast
my $report_obj = Bio::SearchIO->new(-format => 'blast',
                                      -file    => $input);

while( $result_obj = $report_obj->next_result ) {
    while( $hit = $result_obj->next_hit ) {
        while( $hsp = $hit->next_hsp ) {
            if ( $hsp->percent_identity > 60 ) {
                print "Query=", $result_obj->query_name,
                      "\t", "Hit=", $hit->name,
                      "\t", "Length=", $hsp->length('total'),
                      "\t", "Percent_id=", $hsp->percent_identity, "\n";
            }
        }
    }
}
exit;
```

BIOPERL eBLAST

BLAST - Basic Local Alignment and Search Tool

Segundo while:

BIOPERL eBLAST

BLAST - Basic Local Alignment and Search Tool

Segundo while:

```
#!/usr/bin/perl

use strict;
use Bio::Seq;
use Bio::SeqIO;
use Bio::SearchIO;

## Declarando variaveis
my $result_obj; my $hit; my $hsp;

## Argumentos
my $input = $ARGV[0];

## Criando objeto blast
my $report_obj = Bio::SearchIO->new(-format => 'blast',
                                      -file    => $input);

while( $result_obj = $report_obj->next_result ) {
    while( $hit = $result_obj->next_hit ) {
        while( $hsp = $hit->next_hsp ) {
            if ( $hsp->percent_identity > 60 ) {
                print "Query=", $result_obj->query_name,
                      "\t", "Hit=", $hit->name,
                      "\t", "Length=", $hsp->length('total'),
                      "\t", "Percent_id=", $hsp->percent_identity, "\n";
            }
        }
    }
}
exit;
```

BIOPERL eBLAST

BLAST - Basic Local Alignment and Search Tool

Terceiro while:

BIOPERL eBLAST

BLAST - Basic Local Alignment and Search Tool

Terceiro while:

BIOPERL e BLAST

Bio::SearchIO - RESULT METHODS

Object	Method	Example	Description
Result	algorithm	BLASTX	algorithm string
Result	algorithm_version	2.2.4 [Aug-26-2002]	algorithm version
Result	query_name	20521485 dbj AP004641.2	query name
Result	query_accession	AP004641.2	query accession
Result	query_length	3059	query length
Result	query_description	Oryza sativa ... 977CE9AF checksum.	query description
Result	database_name	test.fa	database name
Result	database_letters	1291	number of residues in database
Result	database_entries	5	number of database entries
Result	available_statistics	effectivespaceused ... dbletters	statistics used
Result	available_parameters	gapext matrix allowgaps gapopen	parameters used
Result	num_hits	1	number of hits
Result	hits		List of all Bio::Search::Hit::GenericHit object(s) for this Result
Result	rewind		Reset the internal iterator that dictates where next_hit() is pointing, useful for re-iterating through the list of hits.

BIOPERL e BLAST

Bio::SearchIO - HIT METHODS

Object	Method	Example	Description
Hit	name	443893 124775	hit name
Hit	length	331	Length of the Hit sequence
Hit	accession	443893	accession (usually when this is a genbank formatted id this will be an accession number- the part after the <i>gb</i> or <i>emb</i>)
Hit	description	LaForas sequence	hit description
Hit	algorithm	BLASTX	algorithm
Hit	raw_score	92	hit raw score
Hit	significance	2e-022	hit significance
Hit	bits	92.0	hit bits
Hit	hsps		List of all Bio::Search::HSP::GenericHSP object(s) for this Hit
Hit	num_hsps	1	number of HSPs in hit
Hit	locus	124775	locus name
Hit	accession_number	443893	accession number
Hit	rewind		Resets the internal counter for <code>next_hsp()</code> so that the iterator will begin at the beginning of the list

BIOPERL e BLAST

Bio::SearchIO - HSP METHODS

Object	Method	Example	Description
HSP	evalue	2e-022	e-value
HSP	frac_identical	0.884615384615385	Fraction identical
HSP	frac_conserved	0.923076923076923	fraction conserved (conservative and identical replacements aka "fraction similar") (only valid for Protein)
HSP	gaps	2	number of gaps
HSP	query_string	DMGRCSSG ..	query string from alignment
HSP	hit_string	DIVQNSS ...	hit string from alignment
HSP	homology_string	D+ + SSGCN ...	string from alignment
HSP	length('total')	52	length of HSP (including gaps)
HSP	length('hit')	50	length of hit participating in alignment minus gaps
HSP	length('query')	156	length of query participating in alignment minus gaps
HSP	hsp_length	52	Length of the HSP (including gaps) alias for length('total')
HSP	frame	0	\$hsp->query->frame,\$hsp->hit->frame
HSP	num_conserved	48	number of conserved (conservative replacements, aka "similar") residues
HSP	num_identical	46	number of identical residues
HSP	rank	1	rank of HSP
HSP	seq_inds('query','identical')	(966,971,972,973,974,975 ...)	identical positions as array
HSP	seq_inds('query','conserved-not-identical')	(967,969)	conserved, but not identical positions as array

BIOPERL e BLAST

Bio::SearchIO - HSP METHODS

Object	Method	Example	Description
HSP	seq_inds('query','conserved')	(966,967,969,971,973,974,975, ...)	conserved or identical positions as array
HSP	seq_inds('hit','identical')	(197,202,203,204,205, ...)	identical positions as array
HSP	seq_inds('hit','conserved-not-identical')	(198,200)	conserved not identical positions as array
HSP	seq_inds('hit','conserved',1)	(197,202-246)	conserved or identical positions as array, with runs of consecutive numbers compressed
HSP	score	227	score
HSP	bits	92.0	score in bits
HSP	range('query')	(2896,3051)	start and end as array
HSP	range('hit')	(197,246)	start and end as array
HSP	percent_identity	88.4615384615385	% identical
HSP	strand('hit')	1	strand of the hit
HSP	strand('query')	1	strand of the query
HSP	start('query')	2896	start position from alignment
HSP	end('query')	3051	end position from alignment
HSP	start('hit')	197	start position from alignment
HSP	end('hit')	246	end position from alignment
HSP	matches('hit')	(46,48)	number of identical and conserved as array
HSP	matches('query')	(46,48)	number of identical and conserved as array
HSP	get_aln	<i>sequence alignment</i>	Bio::SimpleAlign object

ChatGPT em Programação

Como usar a IA como aliada no aprendizado e resolução de problemas.

Aula preparada com apoio do ChatGPT.

O que é ChatGPT?

- Uma ferramenta de **Inteligência Artificial** capaz de:
 - Utiliza arquitetura GPT (Generative Pre-trained Transformer)
 - Gerar texto com base em entradas fornecidas.
 - Resolver problemas, sugerir código e explicar conceitos.
- Em programação, é útil para:
 - i. Aprender novos conceitos.
 - ii. Resolver problemas e depurar código.
 - iii. Automatizar tarefas repetitivas.

Exemplos Práticos

1. Ajuda no aprendizado de conceitos

Pergunta:

"Como usar loops em Perl?"

Resposta:

- Explicação sobre `for`, `while`, e `foreach`.
- Exemplos práticos com código.

Exemplos Práticos (continuação)

2. Sugestão e depuração de código

Pergunta:

"Por que este script não está funcionando?"

Resposta:

- ChatGPT pode ajudar a identificar erros e sugerir correções.

Exemplos Práticos (continuação)

3. Automação e produtividade

Pergunta:

"Escreva um script para calcular o GC% de uma sequência de DNA."

Resposta:

- Esboço inicial do script.
- Orientações para personalizar o código.

Limites e Responsabilidades

Nem tudo que o ChatGPT sugere está certo!

- ⚠ Valide o código gerado.
- ⚠ Não dependa exclusivamente da IA.
- ⚠ Use a IA de forma ética.

Conclusão

- O ChatGPT é uma **ferramenta poderosa**, mas **não substitui o aprendizado profundo**.
- Use-o como apoio para explorar novos caminhos.
- Lembre-se: a tecnologia avança, mas o pensamento crítico, a lógica e a criatividade humana continuam sendo as maiores forças no aprendizado e na resolução de problemas.

BOAS FÉRIAS

