# 天津医科大学实验课教案首页



**课程名称：**分子生物计算 **实验名称：**实验 4 基序和循环

**教师姓名：**伊现富 **职称：**讲师 **教学日期：**2015 年 9 月 28 日 10:00-12:00

**授课对象：**生物医学工程与技术学院 2013 级生信班（本） **实验人数：**28

**实验类型（验证型、综合型、设计型、创新型）：**验证型 **实验分组：**一人一机

**学时数：**2 **教材版本：**Perl 语言在生物信息学中的应用——基础篇

**实验目的与要求：**

- 了解 Perl 语言中的流程控制。
- 熟悉 Perl 语言中的 if-elsif-else 和 while；写入文件的方法。
- 掌握 Perl 语言在生物序列数据处理中的应用。

**实验内容及学时分配：**

- (5’) 流程控制：总结 Perl 语言中的条件判断和循环语句。
- (5’) 字符串操作：总结 Perl 语言中常见的字符串操作。
- (5’) 写入文件：总结在 Perl 语言中写入文件的基本步骤。
- (85’) 实验操作：应用 Perl 语言处理生物序列数据。

**主要仪器和实验材料：**

- 主要仪器：一台安装有 Perl 语言（Linux 操作系统）的计算机。

**实验重点、难点及解决策略：**

- 重点难点：应用 Perl 语言处理生物序列数据。
- 解决策略：通过演示进行学习，通过练习熟练掌握。

**思考题：**

- 总结 Perl 语言中的流程控制。
- 总结 Perl 语言中常见的字符串操作。
- 总结 Perl 语言中写入文件的基本步骤。

**参考资料：**


- Beginning Perl for Bioinformatics, James Tisdall, O’Reilly Media, 2001.
- Perl 语言入门（第六版），Randal L. Schwartz, brian d foy & Tom Phoenix 著，盛春 译，东南大学出版社，2012。
- Mastering Perl for Bioinformatics, James Tisdall, O’Reilly Media, 2003.
- 维基百科等网络资源。


**主任签字：** 年 月 日 **教务处制**

一、 流程控制（5 分钟）

1. 条件判断：if，if-else，unless
2. 循环：while，for，foreach

二、 字符串操作（5 分钟）

1. 数组变标量：join
2. 字符串变数组：split
3. 提取子字符串：substr
4. 计数：tr

三、 写入文件（5 分钟）

1. 关联文件和文件句柄
2. 通过文件句柄写入数据
3. 解关联文件和文件句柄

四、 实验操作（85 分钟）

1. if-elsif-else

```
#!/usr/bin/perl -w

$word = 'MNIDDKL';
if ( $word eq 'QSTVSGE' ) {
    print "QSTVSGE\n";
}
elsif ( $word eq 'MRQQDMISHDEL' ) {
    print "MRQQDMISHDEL\n";
}
elsif ( $word eq 'MNIDDKL' ) {
    print "MNIDDKL-the magic word!\n";
}
else {
    print "Is \"$word\" a peptide? This program is not sure.\n";
}
```

2. 在蛋白质序列中查找用户指定的基序

```
#!/usr/bin/perl -w

$proteinfilename = <STDIN>;
chomp $proteinfilename;
unless ( open( PROTEINFILE, $proteinfilename ) ) {
    print "Cannot open file \"$proteinfilename\"\n\n";
    exit;
}
@protein = <PROTEINFILE>;
close PROTEINFILE;

$protein = join( '', @protein );
$protein =~ s/\s//g;
do {
    print "Enter a motif to search for: ";
    $motif = <STDIN>;
    chomp $motif;
    if ( $protein =~ /$motif/ ) {
        print "I found it!\n\n";
    }
    else {
        print "I couldn\'t find it.\n\n";
    }
} until ( $motif =~ /^\s*$/ );
```

3. 对 DNA 序列中的碱基进行计数（策略一）

```
#!/usr/bin/perl -w

print "Please type the filename of the DNA sequence data: ";
$dna_filename = <STDIN>;
chomp $dna_filename;
unless ( open( DNAFILE, $dna_filename ) ) {
    print "Cannot open file \"$dna_filename\"\n\n";
    exit;
}
@DNA = <DNAFILE>;
close DNAFILE;

$DNA = join( '', @DNA );
$DNA =~ s/\s//g;
@DNA = split( '', $DNA );

$count_of_A = 0;
$count_of_C = 0;
$count_of_G = 0;
$count_of_T = 0;
$errors     = 0;

foreach $base (@DNA) {
    if ( $base eq 'A' ) {
        ++$count_of_A;
    }
    elsif ( $base eq 'C' ) {
        ++$count_of_C;
    }
    elsif ( $base eq 'G' ) {
        ++$count_of_G;
    }
    elsif ( $base eq 'T' ) {
        ++$count_of_T;
    }
    else {
        print "!!!!!!!! Error - I don\'t recognize this base: $base
\n";
        ++$errors;
    }
}

print "A = $count_of_A\n";
print "C = $count_of_C\n";
print "G = $count_of_G\n";
print "T = $count_of_T\n";
print "errors = $errors\n";
```

4. 对 DNA 序列中的碱基进行计数（策略二）

```
#!/usr/bin/perl -w

print "Please type the filename of the DNA sequence data: ";
$dna_filename = <STDIN>;
chomp $dna_filename;
unless ( -e $dna_filename ) {
    print "File \"$dna_filename\" doesn\'t seem to exist!!\n";
    exit;
}
unless ( open( DNAFILE, $dna_filename ) ) {
    print "Cannot open file \"$dna_filename\"\n\n";
    exit;
}
@DNA = <DNAFILE>;
close DNAFILE;

$DNA = join( '', @DNA );
$DNA =~ s/\s//g;
$count_of_A = 0;
$count_of_C = 0;
$count_of_G = 0;
$count_of_T = 0;
$errors     = 0;

for ( $position = 0 ; $position < length $DNA ; ++$position ) {
    $base = substr( $DNA, $position, 1 );
    if ( $base eq 'A' ) {
        ++$count_of_A;
    }
    elsif ( $base eq 'C' ) {
        ++$count_of_C;
    }
    elsif ( $base eq 'G' ) {
        ++$count_of_G;
    }
    elsif ( $base eq 'T' ) {
        ++$count_of_T;
    }
    else {
        print "Error - I don\'t recognize this base: $base\n";
        ++$errors;
    }
}
print "A = $count_of_A\n";
print "C = $count_of_C\n";
print "G = $count_of_G\n";
print "T = $count_of_T\n";
print "errors = $errors\n";
```

5. 对DNA序列中的碱基进行计数（策略三），结果保存到文件

```
#!/usr/bin/perl -w

print "Please type the filename of the DNA sequence data: ";
$dna_filename = <STDIN>;
chomp $dna_filename;
unless ( -e $dna_filename ) {
    print "File \"$dna_filename\" doesn\'t seem to exist!!\n";
    exit;
}
unless ( open( DNAFILE, $dna_filename ) ) {
    print "Cannot open file \"$dna_filename\"\n\n";
    exit;
}
@DNA = <DNAFILE>;
close DNAFILE;

$DNA = join( '', @DNA );
$DNA =~ s/\s//g;

$a = 0; $c = 0; $g = 0; $t = 0; $e = 0;

while ( $DNA =~ /a/ig )       { $a++ }
while ( $DNA =~ /c/ig )       { $c++ }
while ( $DNA =~ /g/ig )       { $g++ }
while ( $DNA =~ /t/ig )       { $t++ }
while ( $DNA =~ /[^acgt]/ig ) { $e++ }
print "A=$a C=$c G=$g T=$t errors=$e\n";

$outputfile = "countbase";
unless ( open( COUNTBASE, ">$outputfile" ) ) {
    print "Cannot open file \"$outputfile\" to write to!!\n\n";
    exit;
}
print COUNTBASE "A=$a C=$c G=$g T=$t errors=$e\n";
close(COUNTBASE);
```

6. Perl对数字和字符串的智能化处理

```
#!/usr/bin/perl -w

$num = 1234;
$str = '1234';
print $num, " ", $str, "\n";

$num_or_str = $num + $str;
print $num_or_str, "\n";

$num_or_str = $num . $str;
print $num_or_str, "\n";
```