

# 分子生物计算

## (*Perl* 语言编程)

天津医科大学  
生物医学工程与技术学院

2019-2020 学年上学期 (秋)  
2017 级生信班

# TIMTOWTDI (条条大路通罗马)

伊现富 (Yi Xianfu)

天津医科大学 (TIJMU)  
生物医学工程与技术学院

2019 年 11 月



```
1 # The Perl 5 language interpreter
2 man perl
3 # Look up Perl documentation in Pod format
4 man perldoc
5 perldoc perldoc
6
7 # Perl built-in function
8 perldoc -f BuiltinFunction
9 perldoc -f substr
10 # FAQ
11 perldoc -q FAQKeyword
12 perldoc -q random
13 # Perl predefined variable
14 perldoc -v PerlVariable
15 perldoc -v '$_'
```



# 启用警告

```
1 # Method1
2 # 使用命令行选项-w
3 perl -w script.pl
4
5 # Method2
6 # 在命令解释行使用-w
7 #!/usr/bin/perl -w
8
9 # Method3
10 # 使用use
11 use warnings;
```



# 字符串拼接

```
1 # Method1
2 $DNA3 = "$DNA1$DNA2"; print "$DNA3\n";
3
4 # Method2
5 print "$DNA1$DNA2\n";
6
7 # Method3
8 print $DNA1, $DNA2, "\n";
9
10 # Method4
11 $DNA3 = $DNA1 . $DNA2; print "$DNA3\n";
12
13 # Method5
14 $DNA3 = join "", $DNA1, $DNA2; print "$DNA3\n";
```



# 文件读取

```
1 # Method1 (只读取一行)
2 $protein = <PROTEINFILE>;
3
4 # Method2 (读取所有行, 一次性保存进数组)
5 @proteins = <PROTEINFILE>;
6
7 # Method3 (依次读取每一行)
8 while (<PROTEINFILE>) {
9     ...actions...
10 }
11
12 # Method4 (读取所有行, 一次性保存进标量)
13 $protein = do { local $/; <PROTEINFILE>; };
```



# 获取数组元素个数

```
1 # Methods1
2 $num = scalar @bases;
3
4 # Methods2
5 $num = @bases;
6
7 #Methods3
8 $num = $#bases + 1;
```



# 获取数组的第一个元素

```
1 # Method1
2 $first_gene = $genes[0];
3 #($first_gene) = $genes[0];
4
5 # Method2
6 ($first_gene) = @genes;
7
8 # Method3
9 $first_gene = shift @genes;
10 # 注意: shift会影响原始的@genes数组
```





# 获取数组的最后一个元素

```
1 # Method1
2 $last_gene = $genes[-1];
3
4 # Method2
5 $last_gene = $genes[$#genes];
6
7 # Method3
8 $last_gene = pop @genes;
9 # 注意: pop会影响原始的@genes数组
```



# 变量值 +1

```
1 # Methods1
2 $count++;
3
4 # Methods2
5 ++$count;
6
7 # Methods3
8 $count += 1;
9
10 # Methods4
11 $count = $count + 1;
```



# 计数核苷酸

```
1 # Methods1 (使用数组)
2 @DNA = split( ' ', $DNA );
3 foreach $base (@DNA) { ...count... }
4 # Methods2 (操作字符串)
5 for ( $pos=0 ; $pos < length $DNA ; ++$pos )
6 {
7     $base = substr( $DNA, $pos, 1 ); ..count..
8 }
9 # Methods3 (模式匹配)
10 while ( $DNA =~ /a/ig ) { $a++ }
11 while ( $DNA =~ /[^acgt]/ig ) { $e++ }
12 # Methods4 (使用tr函数)
13 $a = ($DNA =~ tr/Aa/Aa/);
14 $basecount = ($DNA =~ tr/ACGTacgt/ACGTacgt/);
15 $nonbase = (length $DNA) - $basecount;
```





## 考试安排

- 时间：1.15（周二），10:30-12:10
- 地点：西楼 510
- 时长：100 分钟
- 总分：100 分

## 题型与分值

- 1 单选题： $25 \times 1 = 25$
- 2 多选题： $20 \times 1 = 20$
- 3 简答题： $4 \times 5 = 20$
- 4 编程题： $5 \times 5 = 25$
- 5 应用题： $1 \times 10 = 10$

## 考试安排

- 时间：1.15（周二），10:30-12:10
- 地点：西楼 510
- 时长：100 分钟
- 总分：100 分

## 题型与分值

- ① 单选题： $25 \times 1 = 25$
- ② 多选题： $20 \times 1 = 20$
- ③ 简答题： $4 \times 5 = 20$
- ④ 编程题： $5 \times 5 = 25$
- ⑤ 应用题： $1 \times 10 = 10$

- 生物学基础：拉丁语，数据库，碱基/氨基酸代码，限制酶，点突变
- R：常用包
- Markdown：基本语法，格式转换
- Git：基本使用
- Perl 基础：CPAN，变量，赋值，use \*, 三大数据类型，上下文，子程序，调试器，排序，命令行参数，随机，关系数据库，my，编程流程，特殊变量，编程策略，构思步骤，字符串拼接，文件读取写入，perldoc，模块，设计理念，流程控制
- 数组：索引，操作，split, join
- 测试：条件操作符，字符串/数字/文件测试，真假
- 正则表达式：元字符，修饰符，匹配，替换，tr
- 编程：DNA 突变，相似性计算，限制酶的正则表征，子程序传递，序列两两比较，阅读框翻译，……



# FATSQ

1 Read Name → GAGT190851.108399742/1

2 Sequence Bases → TAGATTAGTCTTGTCTTGTCTTCCAGGCTGGAGTCAATGG +

3 '+' →

4 Base Qualities → @004:~57BDABFFA-FFFFFFFFFFH-BECCFFDD-F

1 Read Name → GAGT190851.613197102/1

2 Sequence Bases → CAACATGGTGAACCCGCTCTTCTAAACATACAAATAG +

3 '+' →

4 Base Qualities → CBEBFFIITIGDJIJ177GGHGKGFJETIG111KKKEITK

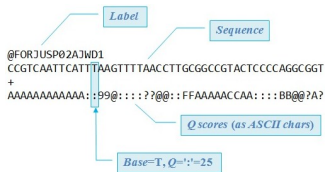
1 Read Name → GAGT190851.22676805/1

2 Sequence Bases → TAGACTAGAGGCTTAAGTCTCAGTCTGGGGCTGTGACATGG +

3 '+' →

4 Base Qualities → @07?CHCEACBEGDEHFDHENGFGHB-GFAEHBEE:EGGT>

Identifier ● @SRX566546.970 HWUSI-EAS1673\_11067\_FC7070M:4:1:2299:1109 length=50  
 Sequence ● TTGCTCGCCTATCATTTTAGTGCTGCTGAGGTGGAGATGTGAGGATCAGT  
 '+' sign ● +  
 Quality scores ● hhhhhhhhhhhghghhhhhhhhhfffef'ee'[X]b[d[ed']Y[Y]  
 Identifier ● @SRX566546.971 HWUSI-EAS1673\_11067\_FC7070M:4:1:2374:1108 length=50  
 Sequence ● GATTTGTATGAAAGTATACAACATAAACTGCAGGTGGATCAGAGTAAGTC  
 '+' sign ● +  
 Quality scores ● hhhhgfhhcghghgfcffdhfhhhhcehdchhdhahehhffde'bVd



A diagram illustrating the components of a FASTQ format line. The line is: `@HWI-EAS3X_10102_2_120_19829_1823#0/2`  
`TCTAACTCTTACTTAGCATAGCTGTAAAAATTTTGAGTT`  
`+(optionally the same identifier)`  
`DEAEE:B:BE5EEED=:DEA:-AE5DDBDFFDEEDFAE`  
Labels with arrows pointing to the corresponding parts of the line:   
- **starting symbol** points to the '@' character.  
- **sequence identifier** points to the text `HWI-EAS3X_10102_2_120_19829_1823#0/2`.  
- **sequence** points to the sequence `TCTAACTCTTACTTAGCATAGCTGTAAAAATTTTGAGTT`.  
- **sequence end start QS** points to the '+' character and the text `(optionally the same identifier)`.  
- **quality score** points to the quality string `DEAEE:B:BE5EEED=:DEA:-AE5DDBDFFDEEDFAE`.

```

! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTU VWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|          |          |          |          |
33         59        64        73         104         126
0.....26...31.....40
SANGER/Illumina 1.8+: Phred+33
-5....0.....9.....40
Solexa: Solexa+64
0.....9.....40
Illumina 1.3+: Phred+64
3.....9.....40
Illumina 1.5+: Phred+64

```







TEX

LATEX

X<sub>Y</sub>TEX

Beamer

