

(共 3 页、第 1 页)

课程名称：分子生物计算

课程内容/章节：基序和循环 / 第 5 章

教师姓名：伊现富

职称：讲师

教学日期：2016 年 11 月 23&30 日 15:30-17:30

授课对象：生物医学工程与技术学院 2013 级生信班（本）

听课人数：30

授课方式：理论讲授

学时数：4

教材版本：Perl 语言在生物信息学中的应用——基础篇

教学目的与要求（分掌握、熟悉、了解、自学四个层次）：

- 掌握：流程控制的语句及语法；文件交互的方法；`join`、`split`、`substr` 和 `tr` 等函数的基本用法；Perl 语言中真与假的判断。
- 熟悉：正则表达式的基本用法；获取键盘输入的方法。
- 了解：常见的特殊变量和文件测试；变量初始化的方法。
- 自学：`join`、`split` 和 `substr` 等函数的高级用法。

授课内容及学时分配:

- (5') 引言与导入：回顾上一章学习的知识点，介绍本章将要学习的内容。
- (35') 流程控制：介绍流程控制的概念和指令，讲解流程控制的分类及各自的语法。
- (10') 代码布局：比较不同的代码布局。
- (50') 查找基序：通过在蛋白质序列中查找用户指定基序的 Perl 程序讲解 Perl 语言的相关知识。
- (70') 计数核苷酸：通过计数核苷酸的 Perl 程序讲解字符串操作等 Perl 语言的相关知识。
- (25') 写入文件：通过保存核苷酸计数结果的 Perl 程序讲解写入文件的方法。
- (5') 总结与答疑：总结授课内容中的知识点与技能，解答学生疑问。

教学重点、难点及解决策略：

- 重点：流程控制的语句及语法；常用字符串操作函数的用法；真与假的判断。
- 难点：Perl 语言中真与假的判断。
- 解决策略：通过实例演示帮助学生理解、记忆。

专业外语词汇或术语:

流程控制 (flow control)

真 (true)

假 (false)

基序 (motif)

初始化 (initialization)

声明 (declare)

辅助教学情况:

- 多媒体：流程控制的各种语法；正则表达式的使用。
- 板书：真与假的判断法则；正则表达式解析。
- 演示：常见字符串处理函数的使用。

复习思考题:

- 总结 Perl 语言中流程控制的语句及其语法。
- 总结 Perl 语言真与假的判断法则。
- 总结字符串和数组互转的方法。
- 总结正则表达式和模式匹配的使用。
- 总结 Perl 对数字和字符串的智能化处理。
- 总结 substr 和 tr 处理字符串的用法。

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、 引言与导入 (5 分钟)

1. 已经学习

- Perl 语言基础: 标量、数组、字符串操作、读取文件
- 生物序列处理: 拼接 DNA 片段、DNA 转录成 RNA、获取反向互补序列

2. 即将学习

- Perl 语言基础: 条件测试、循环、正则表达式、写入文件
- 生物序列处理: 查找基序、计数核苷酸

二、 流程控制 (35 分钟)

1. 简介

(1) 定义: 在程序运行时, 个别的指令 (或是陈述、子程序) 运行或求值的顺序

(2) 指令

- 继续运行位于不同位置的一段指令
- 若特定条件成立时, 运行一段指令
- 运行一段指令若干次, 直到特定条件成立为止
- 运行位于不同位置的一段指令, 但完成后会继续运行原来要运行的指令
- 停止程序, 不运行任何指令

2. 【重点】分类

(1) 默认: 除非明确指明不按顺序执行, 否则程序将从最顶端的第一个语句开始, 顺序执行到最底端的最后一个语句

(2) 条件判断: 只在条件测试成功的前提下执行相应的语句, 否则直接跳过这些语句; if、if-else、unless (通过实例详细讲解各自的语法, 比较常规写法和简写, 比较 if 和 unless)

- Perl 程序 5.1: 使用 if-elsif-else

```
if( 1 == 1 ) {                                if( 1 == 0 ) {
print "1 equals 1\n";                          print "1 equals 0\n";
}                                                }
if( 1 ) {                                       if( 0 ) {
print "1 evaluates to true\n";                 print "0 evaluates to true\n";
}                                                }
```

(3) 循环: 一直重复语句, 直到相应的测试失败为止; while、for、foreach

- Perl 程序 5.2: 使用 while 从文件中读取蛋白质序列数据

3. 【重点、难点】真和假

(1) 数字: 0 为假, 其他为真

(2) 字符串: 空字符串和字符串 '0' (唯一为假的非空字符串) 为假, 其他为真

(3) 其他: 先转换成数字或字符串再行判断

三、 代码布局 (10 分钟) (比较不同代码布局的优缺点)

```
while ( $alive ) {                            while ( $alive ) {
    if ( $needs_nutrients ) {                  {
        print "Needs nutrients\n";              if ( $needs_nutrients )
    }                                           {
}                                               print "Needs nutrients\n";
}                                               }
}                                               }
```

四、 查找基序 (50 分钟)

1. 问题分析与解决策略：基序长度不定有变体；正则表达式
2. Perl 程序 5.3：在蛋白质序列中查找用户指定的基序
3. 获取键盘输入：<STDIN>，chomp vs. chop
4. **【重点】** 数组变量：\$protein = join ('', @protein); (拼接两个 DNA 片段的又一种方法)
5. do-until：先执行后测试 (与 until 进行比较)
6. 正则表达式：\$protein =~ s/\s//g; \$motif =~ /^s*\$/ (结合实例解释其中的字符集、元字符、量词、锚位等)
7. 模式匹配：\$protein =~ /\$motif/ (比较变量内插和直接使用字符串的优缺点)

五、 计数核苷酸 (70 分钟) (同样的碱基数，不同的策略方法，各有优势与劣势)

1. 伪代码 (通过简洁的伪代码理清思路)
2. 策略 (思考每种策略的优缺点)
 - (1) 把 DNA 拆解成单个碱基，存储到数组中，对数组中的元素进行迭代处理
 - (2) 对 DNA 字符串中的位置进行迭代处理
3. 把字符串拆解成数组
 - (1) Perl 程序 5.4：对 DNA 序列中的碱基进行计数
 - (2) **【重点】** split: @DNA = split(' ', \$DNA); (和 join 进行比较)
 - (3) 初始化
 - 未初始化变量的值为 'undef'——0 或者空字符串 (Perl 中的上下文无处不在)
 - Perl 程序 5.5：Perl 对数字和字符串的智能化处理
 - (4) foreach: foreach \$base (@DNA) { (比较自定义变量和内置变量的利弊)
 - (5) +1: 至少四种实现方法 (条条大路通罗马，比较各种方法的优缺点)
4. 操作字符串
 - (1) Perl 程序 5.6：对 DNA 序列中的碱基进行计数
 - (2) 文件测试：unless, -e
 - (3) for vs. foreach (语法不同，本质一样)
 - (4) 索引：不管是字符串还是数组元素，都从 0 开始索引
 - (5) **【重点】** substr: \$base = substr(\$DNA, \$position, 1); (通过实例详细讲解其用法；substr vs. splice——操作字符串 vs. 操作数组)

六、 写入文件 (25 分钟)

1. Perl 程序 5.7：对 DNA 序列中的碱基进行计数，结果保存到文件
2. 读取文件 vs. 写入文件
3. while：妙用正则匹配进行计数
4. **【重点】** tr：妙用 tr 进行计数，优劣并存

七、 总结与答疑 (5 分钟)

1. 知识点
 - 流程控制：条件、循环
 - 文件交互：打开、读取、写入
 - 正则匹配：正则表达式，模式匹配
 - 字符串操作：join、split、substr、tr
 - 其他：真与假、获取键盘输入、变量递增、文件测试、……
2. 技能
 - 熟练使用 Perl 语言中的各种流程控制语句
 - 能够编写在 DNA 或者蛋白质序列中查找基序的程序
 - 能编写对 DNA 序列中的核苷酸进行计数的程序