# 天津医科大学理论课教案首页

(共3页、第1页)

课程名称:分子生物计算 课程内容/章节:子程序和 Bugs / 第 6 章

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: 子程序的使用——定义、调用、数据传递等。

• 熟悉: Perl 调试器的基本用法。 • 了解: 模块的编写与使用。

• 自学: Perl 调试器的高级用法。

#### 授课内容及学时分配:

• (5') 引言与导入: 对子程序和 Perl 调试器进行简介。

- (30') 子程序: 讲解子程序的基本用法,包括定义、调用、参数、作用域等。
- (15') 命令行参数和数组: 讲解命令行参数的使用, 以及相关的特殊变量。
- (50') 传递数据给子程序: 讲解传递数据给子程序的方法, 对值传递和引用传递进行比较。
- (10') 模块和子程序库:介绍模块的编写和使用。
- (85') 修复 Bugs: 总结调试程序的方法, 讲解 Perl 调试器的使用。
- (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: 子程序的基本用法; 传递数据给子程序的方法。

• 难点:传递数据给子程序的方法; Perl 调试器的使用。

• 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:

子程序 (subroutine) 调试器 (debugger) 程序错误 (bug)

参数 (argument/parameter)

值 (value)

引用 (reference) 模块 (module) 库 (library)

### 辅助教学情况:

• 多媒体: 子程序的基本用法; 模块的使用。

• 板书: 传递数据给子程序的方法。

• 演示: Perl 调试器的使用。

### 复习思考题:

• 如何定义和调用子程序?

• 举例说明作用域的概念。

• 如何获取命令行参数?

- 举例说明传递数据给子程序的方法。
- 总结调试 Perl 程序的方法。
- 如何使用 Perl 调试器调试程序?

## 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

# 天津医科大学理论课教案续页

(共3页、第2页)

- 一、 引言与导入 (5分钟)
  - 子程序: 对程序进行结构化组织
  - Perl 调试器: 使用"慢镜头"检查程序的行为
- 【重点】子程序(30分钟)
  - 1. 简介
    - 简介: 程序中的程序 ⇒ 包裹代码、起个名字、传递数据、运算返回结果
    - 优势:一次编写、多次使用 ⇒ 简短易理解、稳健易测试、灵活模块化
    - 原则: 只做一件事情并把它做好 ⇒ 通用且有用、不止使用一次、代码不超过一页
  - 2. 编写(通过实例演示讲解)
    - 程序分块: 主程序(从开头到 exit 命令) + 子程序(其余部分)
    - 定义与调用: 集中定义, 通过子程序名(和参数)进行调用
  - 3. 定义
    - 语法: sub + 名字 + 代码块
    - 两类变量:传递给子程序的参数,子程 \$dna .= 'ACGT'; 序中声明的变量
    - 返回值: return, 标量、列表、数组等 }
  - 4. 参数
    - 传递: 变量名无关紧要, 关键是值及其顺序
    - 收集: 特殊变量 @ 数组
  - 5. 作用域(比较使用 my 前后程序的输出)
    - 作用域: 把变量隐藏起来, 使它们仅局限在程序的特定部分
    - 词法作用域: mv, 把变量限制在使用它们的代码块中
- 三、 命令行参数和数组 (15 分钟)
  - 1. 命令行参数
    - 非交互、自动化
    - \$0: 程序名
    - @ARGV: 所有的命令行参数
- 2. 数组使用

sub addACGT {

my (\$dna) = 0;

return \$dna;

- 起始索引为 0
- 使用 \$ 而非 @ 提取元素
- 在中括号中放置下标
- 【重点、难点】传递数据给子程序(50分钟)(通过实例进行演示,并对两者进行比较) 四、
  - 1. 通过值传递
    - 实例: simple\_sub(\$i);
    - 说明
      - 调用子程序时参数的值被复制并传递给子程序
      - 子程序中值的变化不会影响到主程序中相应参数的值
    - 适用: 单个标量、标量列表、单个数组等
  - 2. 通过引用传递
    - 实例: reference sub(\@i, \@j); (vs. reference\_sub(@i, @j);
    - 说明
      - 引用: 在变量名前加\, 存储在标量变量中的一种特殊类型的数据
      - 收集: 从 @ 中读取参数时要保存为标量变量
      - 使用: 先解引用后使用
      - 解引用: 在引用前添加表明变量类型的符号(\$、@、8)
      - 注意: 子程序中对参数变量值的操作会影响到主程序中参数的值
    - 适用:复杂参数(混合标量、数组和散列)

# 天津医科大学理论课教案续页

(共3页、第3页)

## 五、 模块和子程序库 (10分钟)

- 1. 模块
  - 作用:避免繁琐、重复地复制粘贴子程序
  - 收集: 把常用的子程序统一放在一个文件中
  - 模块: 最后一行必须是 1; 、后缀为 .pm
  - 使用: 在程序顶部使用 use 语句, 不需要后缀
  - 实例: use BeginPerlBioinfo;
- 2. 子程序库
  - 作用: 指定保存模块的绝对路径
  - 实例: use lib '/home';

## 六、 修复 Bugs (85 分钟)

- 1. bug
  - 简介:程序设计术语,程序错误/漏洞,源于历史中的典故
  - 常见: 括号没配对、没用分号结尾、拼写错误、索引错误、……
- 2. use warnings; 和use strict;
  - use warnings;: 开启警告功能
  - use strict;: 强制使用 my 声明变量……
- 3. 使用注释和 print 语句
  - 选择性注释:适用于没有精确定位错误、但是直到大体范围时
  - 添加 print 语句: 打印出变量的值,适用于差不多直到错误之处时
- 4. 【难点】Perl 调试器 (通过实例演示其使用)
  - 启动与停止: perl -d script.pl; 输入 q 退出
  - 使用帮助: man perldebug; h; h h; h CMD
  - 常用命令: p、n、s、v、b、c、R、……

#### 七、 总结与答疑 (5 分钟)

- 1. 知识点
  - 子程序: 定义,调用,返回值,参数,作用域
  - 命令行参数: 特殊变量, 提取数组元素
  - 传递数据给子程序: 通过值 vs. 通过引用
  - 模块:编写、使用,指定库目录
  - 调试: use warnings; 和use strict;, 注释和 print, 调试器
  - Perl 调试器: 启动和停止, 常用命令
- 2. 技能
  - 能够熟练使用子程序
  - 能够调试 Perl 程序
  - 能够熟练使用 Perl 调试器