

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：编程的艺术 / 第 3 章

教师姓名：伊现富 职称：讲师 教学日期：2018 年 11 月 19 日 10:00-11:40

授课对象：生物医学工程与技术学院 2016 级生信班（本） 听课人数：28

授课方式：理论讲授 学时数：2 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：编程的策略、步骤和基本流程；调试程序的主要方法。
- 熟悉：学习编程的主要方法；伪代码的编写与解读。
- 了解：Git 在版本控制中的应用。
- 自学：GitHub 的使用。

授课内容及学时分配：

- (5') 引言与导入：回顾学习 C 语言和 Linux 的经历。
- (10') 学习方法：总结学习编程的各种方法。
- (30') 编程流程：介绍编程的基本流程，介绍版本控制的概念，详细讲解 Git 的使用。
- (10') 编程策略：总结编程的主要策略。
- (30') 编程过程：通过实例讲解编程的基本步骤，介绍伪代码和代码注释。
- (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：编程的策略、步骤和流程。
- 难点：编程的步骤和流程。
- 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：

版本控制系统 (VCS, Version Control System)	伪代码 (pseudocode)
算法 (algorithm)	注释 (comment)

辅助教学情况：

- 多媒体：编程中的版本控制；伪代码。
- 板书：编程的基本流程；解释算法的实例。
- 演示：Git 的使用。

复习思考题：

- | | |
|--------------------|-------------------|
| • 总结学习编程语言的主要方法。 | • 总结常用的编程策略。 |
| • 编写程序的基本流程是什么？ | • 编程的基本过程是怎么样的？ |
| • 如何使用 Git 进行版本控制。 | • 在编程过程中需要构思哪些内容？ |
| • 总结调试程序的方法。 | • 使用伪代码和注释有哪些优势？ |

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) (引导学生进行回顾)

- C 语言的学习：听课、编程、练习……
- Linux 系统的学习：听课、自学、练习……

二、 学习方法 (10 分钟) (通过自己的亲身经历进行讲解)

1. 最佳方法：适合自己的才是最好的！ (取决于任务属性、时间限制、个人喜好……)
2. 常见方法：参加培训班、阅读书籍、死啃手册、拜师学艺、研读程序……
3. 五字真言：实践出真知！ (Experience is the best teacher.)

三、 【重点、难点】编程流程 (30 分钟) (通过实例进行讲解)

1. 基本流程：编辑-运行-修正
2. 版本控制

(1) 简介：一种记录若干文件内容变化，以便将来查阅特定版本修订情况的系统

(2) Git

- 简介：分散式版本控制软件，Linus Torvalds，2005 年
- 使用 (通过简单的实例演示使用方法)

(3) GitHub：一个共享虚拟主机服务，用于存放使用 Git 版本控制的软件代码和内容项目；世界上最大的代码存放网站

3. 错误信息

- 出现错误并不可怕
- 一定不要视而不见
- 从第一个错误开始修复
- 必要时进行一定的猜测

4. 调试程序

- Perl 调试器：perl -d script.pl
- 加入 print 语句
- 选择性得注释掉部分代码
- 模 块：Benchmark, Data::Dumper, Smart::Comments, ...

Basic Git Workflow Example

Initialize a new git repository, then stage all the files in the directory and finally commit the initial snapshot.

```
$ git init
$ git add .
$ git commit -m 'initial commit'
```

Create a new branch named featureA, then check it out so it is the active branch. then edit and stage some files and finally commit the new snapshot.

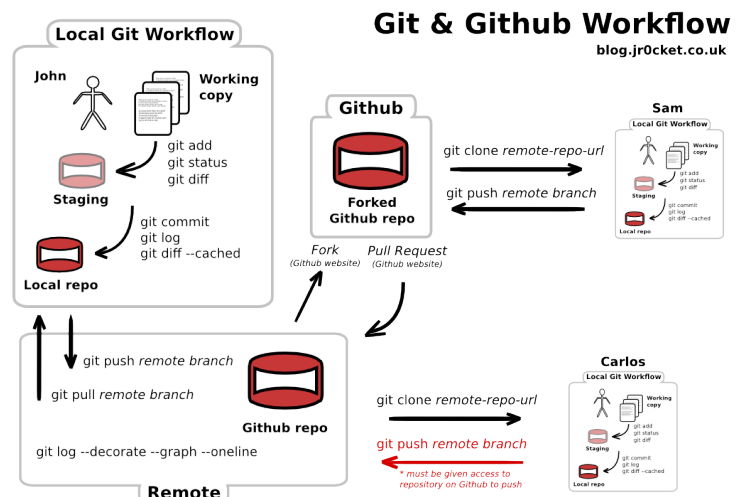
```
$ git branch featureA
$ git checkout featureA
$ (edit files)
$ git add (files)
$ git commit -m 'add feature A'
```

Switch back to the master branch, reverting the featureA changes you just made, then edit some files and commit your new changes directly in the master branch context.

```
$ git checkout master
$ (edit files)
$ git commit -a -m 'change files'
```

Merge the featureA changes into the master branch context, combining all your work. Finally delete the featureA branch.

```
$ git merge featureA
$ git branch -d featureA
```



四、 【重点】编程策略 (10 分钟) (通过现实中的例子进行讲解)

1. 寻找现成的程序 (避免重复发明轮子)
2. 自己编写程序
 - (1) 修改现成的程序 (注意：有时并不比从头编程容易！)
 - (2) 充分利用已有的模块快速拼凑程序
 - (3) 从头编写完整的程序
3. 请其他专家编写程序

五、【重点、难点】编程过程 (30 分钟) (以计算一个 DNA 序列中调控元件的数目为例进行讲解)

1. 基本步骤: 确定输入 \Rightarrow 整体构思 \Rightarrow 确定输出 \Rightarrow 改善构思 \Rightarrow 编写程序
2. 程序构思: 确定输入输出, 选择算法与数据结构, 选择编程范式, …… (三思而后行)
3. 算法: 计算的思路与具体步骤 (类似于现实生活中处理问题的步骤)
4. 伪代码: 将整个算法运行过程的结构用接近自然语言的形式描述出来 (介于编程语言和自然语言的“中间语言”)

```
get the name of DNAfile
```

```
read in the DNA from the DNAfile
```

```
for each regulatory element
```

```
    if element is in DNA, then
```

```
        add one to the count
```

```
print count
```

5. 注释 (牢记: 程序不仅是给计算机看的, 也会被人查看)

- 以# 进行注释
- 目的、思路、输入输出、……
- 妙用: 在源代码中保留伪代码

六、总结与答疑 (5 分钟)

1. 知识点

- 学习编程: 培训班、书籍、手册、……
- 基本流程: 编辑-运行-修正
- 版本控制: Git, GitHub
- 程序调试: 调试器, print, 注释, 模块, ……
- 编程策略: 找现成程序, 自己编写, 找人代写, ……
- 基本步骤: 构思 (输入输出, 算法), 编程 (伪代码, 注释)

2. 技能

- 熟练应用编程的基本策略、步骤和流程
- 能够使用 Git 和 GitHub 进行版本控制
- 能够使用不同的方法调试程序