

天津医科大学理论课教案首页

(共 5 页、第 1 页)

课程名称：分子生物计算 课程内容/章节：限制酶图谱和正则表达式 / 第 9 章

教师姓名：伊现富 职称：讲师 教学日期：2016 年 12 月 28 日 13:30-17:30

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：正则表达式的基础语法与基本应用；模式匹配中的特殊变量。
- 熟悉：逻辑操作符的求值顺序。
- 了解：范围操作符的应用。
- 自学：操作符的优先级。

授课内容及学时分配：

- (5') 引言与导入：回顾已经学习的与正则表达式和操作符相关的知识点；介绍将要学习的主要内容。
- (80') 正则表达式：通过实例介绍正则表达式的应用，详细讲解正则表达式的常量、运算、语法和元字符等基础知识，举例说明正则表达式在生物信息学中的应用。
- (100') 限制酶切图谱：简单介绍限制酶的背景知识，通过制作酶切图谱的 Perl 程序详细讲解正则表达式在限制酶切位点分析中的应用。
- (10') 操作符优先级：简单介绍优先级的概念，总结常见操作符的优先级。
- (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：正则表达式的基本语法；正则表达式中的元字符；模式匹配中的特殊变量。
- 难点：正则表达式的基本语法；逻辑操作符的求值顺序；pos 函数的使用。
- 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：

正则表达式 (regular expression)	限制酶 (restriction enzyme)
模式 (pattern)	回文序列 (palindrome sequence)
元字符 (metacharacter)	范围操作符 (range operator)
优先级 (precedence)	逻辑操作符 (logical operator)

辅助教学情况：

- 多媒体：正则表达式实例；限制酶酶切位点示意图；bionet 文件格式。
- 板书：逻辑操作符的求值顺序；使用括号明确优先级。
- 演示：正则表达式在酶切图谱制作中的应用。

复习思考题：

- 总结正则表达式的基本运算。
- 总结正则表达式的基本语法。
- 举例说明正则表达式中的元字符。
- 解析正则表达式实例。
- 根据要求编写正则表达式。
- 列举常见的逻辑操作符，解释其求值顺序。
- 举例说明模式匹配中的特殊变量。
- 如何明确复杂表达式中操作的优先级？

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：模式匹配与字符替换（正则表达式）；数字和字符操作符（操作符）
- 生物信息学：处理 FASTA 格式的文件；在 DNA 序列中查找基序

2. 即将学习

- Perl 语言：正则表达式的基本理论；操作符的优先级
- 生物信息学：用正则表达式表征酶切数据；制作酶切图谱

二、正则表达式 (80 分钟)

1. 简介 (日用而不知：文本编辑器中的检索和替换)

正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。

2. 实例 (DNA 基序、文本搜索、用户名、电子邮箱、网址等)

- /CT[CGT]ACG/: 完整的正则表示式 (基序)
- //: 正则表达式界定符
- ACGT: ACGT 四种碱基/四个字符本身
- [CGT]: C 或者 G 或者 T
- 基序: CTCACG 或者 CTGACG 或者 CT-TACG

/([il]f|and)* [AC]+.(and)?/

"You might," the candid hero admitted, "though such obtuse...
...id certainly be phenomenal. Still, the event is possible. So I
...st ask you to grant one more Hypothetical."
"Very good. I'm quite willing to grant it, as soon as you've written
...t down. We will call it
(D) If A and B and C are true, Z must be true.
"Have you entered that in your note-book?"
"I have!" Achilles joyfully exclaimed, as he ran the pencil into its
sheath. "And at last we've got to the end of the ideal race-course!
Now that you accept A and B and C and D, of course you
accept Z."
"Do I?" said the Tortoise innocently. "Let's make that quite clear.
I accept A and B and C and D. Suppose I still refuse to accept
Z?"
"Then Logic would take you by the throat, and force you to do it!"
Achilles triumphantly replied. "Logic would tell you 'You can't help
yourself. Now that you've accepted A and B and C and D, you
must accept Z'. So you've no choice, you see."

3. 理论

(1) 基本理论

- 简介：常量 (字符串的集合) + 算子 (集合上的运算)
- 常量：空集，空串，文字字符
- 运算：串接，选择，Kleene 星号
- 运算优先级：Kleene 星号 > 串接 > 选择

(2) 基本语法

- 简介：一个正则表达式通常被称为一个模式，为用来描述或者匹配一系列符合某个句法规则的字符串。
- 【重点、难点】语法 (结合实例讲解)
 - 选择：|, /gray|grey/
 - 数量限定：+, ?, *
 - 匹配：(), 定义操作符的范围和优先级
- 【重点】元字符 (结合实例讲解)
 - 简介：一个或一组代替一个或多个字符的字符
 - 元字符集
 - 元字符优先级

a (b c)	= {ab, ac}
(a b) (c d)	= {ac, ad, bc, bd}
aa*	= a ⁺ = {a, aa, aaa, ...}

字符	描述
\	将下一个字符标记为一个特殊字符、或一个原义字符、或一个向后引用、或一个八进制转义符。例如，“n”匹配字符“n”。“\n”匹配一个换行符。序列“\\”匹配“\”而“\”则匹配“\”。
^	匹配输入字符串的开始位置。如果设置了RegExp对象的Multiline属性，^也匹配“\n”或“\r”之后的位置。
\$	匹配输入字符串的结束位置。如果设置了RegExp对象的Multiline属性，\$也匹配“\n”或“\r”之前的位置。
*	匹配前面的子表达式零次或多次。例如，“zo*”能匹配“z”、“zo”以及“zoo”。*等价于{0,}。
+	匹配前面的子表达式一次或多次。例如，“zo+”能匹配“zo”以及“zoo”，但不能匹配“z”。+等价于{1,}。
?	匹配前面的子表达式零次或一次。例如，“do(es)?”可以匹配“do”或“does”中的“do”。?等价于{0,1}。
{n}	n是一个非负整数。匹配确定的n次。例如，“o{2}”不能匹配“Bob”中的“o”，但是能匹配“food”中的两个o。
{n,}	n是一个非负整数。至少匹配n次。例如，“o{2,}”不能匹配“Bob”中的“o”，但是能匹配“fooooo”中的所有o。“o{1,}”等价于“o+”。“o{0,}”则等价于“o*”。
{n,m}	m和n均为非负整数，其中n≤m。最少匹配n次且最多匹配m次。例如，“o{1,3}”将匹配“foooooo”中的前三个o。“o{0,1}”等价于“o?”。请注意在逗号和两个数之间不能有空格。
?	当该字符紧跟在任何一个其他限制符(*,+,?,{n},{n,},{n,m})后面时，匹配模式是非贪婪的。非贪婪模式尽可能少的匹配所搜索的字符串，而默认的贪婪模式则尽可能多的匹配所搜索的字符串。例如，对于字符串“oooo”，“o+?”将匹配单个“o”，而“o+”将匹配所有“o”。
.	匹配除“\n”之外的任何单个字符。要匹配包括“\n”在内的任何字符，请使用像“(. \n)”的模式。

优先权	符号
最高	\
高	(), (? :), (?=), []
中	*, +, ?, {n}, {n,}, {m,n}
低	^, \$, 中介字符
最低	

天津医科大学理论课教案续页

(共5页、第3页)

4. 生物学应用 (综合运用前述理论知识，讲解正则表达式在专业中的应用)

- 匹配 1~6 号染色体
- 匹配任意一个碱基/核苷酸
- *Bst*YI 的切割序列 (RGATCY)
- <A-X-[ST](2)-X(0,1)-{V}
- /chr[1-6]/
- /[ACGTU]/
- /[AG]GATC[TC]/
- /^A.[ST]{2}.?[^V]/

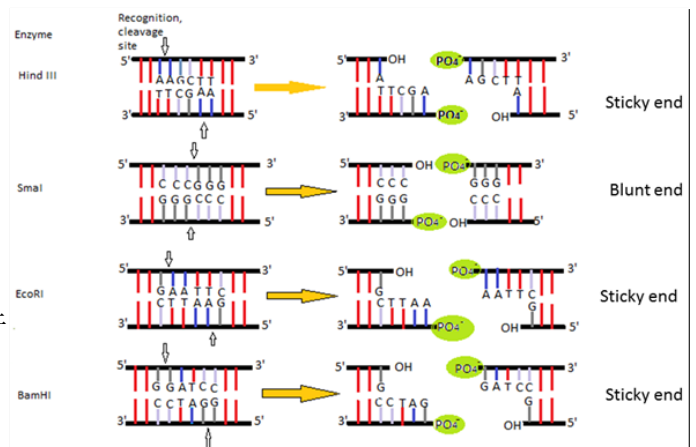
三、限制酶切图谱 (100 分钟)

1. 限制酶

- 限制酶：切割双链 DNA，黏性末端或者平滑末端
- 分类：Type I, Type II (回文序列), Type III

2. 程序规划

- 目的：制作 DNA 序列酶切图谱
- 输入
 - DNA 序列：读取 FASTA 文件
 - 限制酶数据：REBASE
- 处理
 - 表征：限制酶 ⇒ 正则表达式
 - 存储：散列 (酶的名字 ⇒ 酶切位点)
 - 查询：向用户询问酶的名字
- 输出：酶的名字，位置列表
- 总结
 - 限制酶翻译成正则表达式 [?]
 - 把限制酶存储在散列中 [!]
 - 从 FASTA 文件中读入 DNA 序列 [!]



3. 限制酶数据

- 数据来源：REBASE 数据库 ⇒ bionet 格式的文件
- 知识点：split (处理特殊变量 \$_) ; shift/pop (提取数组的第一个/最后一个元素)
- Perl 程序 9.1：把 IUB 核酸代码转换成正则表达式
- Perl 程序 9.2：解析 REBASE 中 bionet 格式的数据文件

4. 操作符

- 范围操作符：..
- 逻辑操作符：and, or, not
- 【难点】逻辑操作符的求值顺序

- **and**: 左边为真时, 对右边求值返回结果; 左边为假时, 直接返回结果, 右边永远不会被求值
- **or**: 左边为假时, 对右边求值返回结果; 左边为真时, 直接返回结果, 右边永远不会被求值

5. 制作酶切图谱

- **【重点】** 特殊变量 (原字符串 = `$` + $& + $'`) (**通过实例进行讲解**)
 - `$``: 实际匹配模式之前的部分
 - `$&`: 实际匹配模式的部分
 - `$'`: 实际匹配模式之后的部分
- **【难点】** `pos` 函数 (**通过实例进行讲解; 注意索引从 0 开始**)
 - `pos`: 返回匹配序列后面第一个字符的索引位置
 - `pos-length`: 返回匹配序列第一个字符的索引位置
- Perl 程序 9.3: 根据用户输入的酶的名字制作酶切图谱

四、操作符优先级 (10 分钟)

1. 优先级：操作符操作顺序的规则 (普通会员 vs. 白金会员 vs. 钻石会员)
2. 基本原则：使用括号明确操作顺序

五、总结与答疑 (5 分钟)

1. 知识点

- 正则表达式：基础（理论、语法、元字符等）；应用（解析、构建）
- 操作符：范围操作符；逻辑操作符（求值顺序）；优先级
- 模式匹配：特殊变量，pos 函数

2. 技能

- 能够把 IUB 代码翻译成正则表达式
- 能够编写制作酶切图谱相关的 Perl 程序