

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

第一章 绪论

伊现富 (Yi Xianfu)

天津医科大学 (TJMU)
生物医学工程与技术学院

2019 年 8 月



教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题

教学提纲

1 课程安排

2 生物信息学

3 生物学

4 计算机科学

5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题





前 9 周，每周三/五，
上午后两节（10:00-11:40）/下午前两节（13:30-15:10），
西楼 501/510

授课内容

- 教材内容：第一章……～第十章
- 补充知识：Markdown, Git, ...



- 上课期间以小组形式（建议以宿舍为单位）编写程序
- 任务主题不限（兴趣主导，生信方面的最好）
- 理论课的最后一/两次课（2~4 学时）进行报告
- 小组成员人人都要参与
- 既参与编程又进行报告者加分
- 作为平时成绩的一大部分



目的

按照用户的需求生成随机密码。

需求

- 指定生成密码的元素（默认：混合使用数字、大小写字母、符号）
- 指定生成密码的长度（默认：12 个字符）
- 指定生成密码的个数（默认：1 个）
- 指定密码中元素的比例/字符数（默认：无比例/字符数要求）
- 指定需要排除的元素（比如：0 和 O, 1 和 I；默认：无）
- 挖掘更多人性化需求……

代码仓库

- [Yixf-Education/project_Perl](#)

目的

按照用户的需求生成随机密码。

需求

- 指定生成密码的元素（默认：混合使用数字、大小写字母、符号）
- 指定生成密码的长度（默认：12个字符）
- 指定生成密码的个数（默认：1个）
- 指定密码中元素的比例/字符数（默认：无比例/字符数要求）
- 指定需要排除的元素（比如：0和O, 1和I；默认：无）
- 挖掘更多人性化需求……

代码仓库

- [Yixf-Education/project_Perl](#)

目的

按照用户的需求生成随机密码。

需求

- 指定生成密码的元素（默认：混合使用数字、大小写字母、符号）
- 指定生成密码的长度（默认：12个字符）
- 指定生成密码的个数（默认：1个）
- 指定密码中元素的比例/字符数（默认：无比例/字符数要求）
- 指定需要排除的元素（比如：0和O, 1和I；默认：无）
- 挖掘更多人性化需求……

代码仓库

- Yixf-Education/project_Perl

前 9 周，每周三，下午前两节（13:30-15:10），教一楼 304

实验内容

- 紧跟理论课进度
- 幻灯片与教材上的程序/实例



① 理论课：80%

- ① 平时表现：5%
- ② 课堂测验：5%（期中）+5%（期末）
- ③ 自主学习：15%
- ④ 闭卷考试：50%

② 实验课：20%

- ① 平时表现：10%
- ② 实验报告：10%



教学提纲

1 课程安排

2 生物信息学

3 生物学

4 计算机科学

5 编程

6 Bioinformatics Career Survey
2008

7 R

8 书籍

9 回顾与总结

- 总结

- 思考题



生物信息学 (bioinformatics)

"If you can't do bioinformatics, you can't do biology."

Lincoln Stein (Biologist and former CSHL Professor)

```
CCGCT);GTATTCGTACATTACTGCCAGCCACCATGAATATTGTACGGTACC
A#print STDERR 'blast args: ', Dumper( \@args ), $/;A
CAACCACTAGGATAACCAAAACCTACCCACCCCTAACAGTACATAGTACATC
C# my $pcf = DNALC::Pipeline::Config->new->cf('PIPELINE'
Cmy $blast_script = File::Spec->catfile($pcf->{EXE_PA
Amy $rc = system($blast_script, @args);CAATCAACCCCTATA
Cprint STDERR "blast rc = $rc\n";TTAACAGTACATAGTACATC
CTGTTCTTCATGGGGAAAGCAGATTGGGTACCACCCAAAGTATTGACCACCCA
C# 0 == successTACATTACTGCCAGCCACCATGAATATTGTACGGTACC
A# 2 == success, no resultsAAGCAAGTACAGCAATCAACCCCTATA
Cif ((0 == $rc || 2 == $rc) && -f $out_file) {GTACATC
CTGTTmy $alignment = '';TTGGGTACCACCCAAAGTATTGACCACCCA
CCGCTif ($fh->open($out_file)) {CCATGAATATTGTACGGTACC
ATATCAAAwhile (<$fh>) {TACAAGCAAGTACAGCAATCAACCCCTATA
CACCACTAGGAT$alignment .= $_;CCCTTAACAGTACATAGTACATC
CTGTTCTT)ATGGGGAAAGCAGATTGGGTACCACCCAAAGTATTGACCACCCA
CCGCTATGT$fh->close;TACTGCCAGCCACCATGAATATTGTACGGTACC
ATATC)AAACCCCCCTCCCCATGCTTACAAGCAAGTACAGCAATCAACCCCTATA
CACCC$blast = DNALC::Pipeline::Phylogenetics::Blast->
CACCACTAGGATproject_id => $self->project->id,GTACATC
```

Nothing in Biology Makes Sense Except in the Light of Evolution
and Bioinformatics.

生物信息学（bioinformatics）利用应用数学、信息学、统计学和计算机科学的方法研究生物学的问题。

生物信息学的**研究材料和结果**就是各种各样的生物学数据，其**研究工具**是计算机，**研究方法**包括对生物学数据的搜索（收集和筛选）、处理（编辑、整理、管理和显示）及利用（计算、模拟）。

目前主要的研究方向有：序列比对、序列组装、基因识别、基因重组、蛋白质结构预测、基因表达、蛋白质反应的预测，以及建立进化模型。



生物学技术往往生成大量的嘈杂数据。与数据挖掘类似，生物信息学利用数学工具从大量数据中提取有用的生物学信息。生物信息学所要处理的典型问题包括：重新组装在霰弹枪测序法测序过程中被打散的 DNA 序列，从蛋白质的氨基酸序列预测蛋白质结构，利用 mRNA 微阵列或质谱仪的数据检验基因调控的假说。

某些人将计算生物学作为生物信息学的同义词处理；但是另外一些人认为计算生物学和生物信息学应当被当作不同的条目处理，因为生物信息学更侧重于生物学领域中计算方法的使用和发展，而计算生物学强调应用信息学技术对生物学领域中的假说进行检验，并尝试发展新的理论。



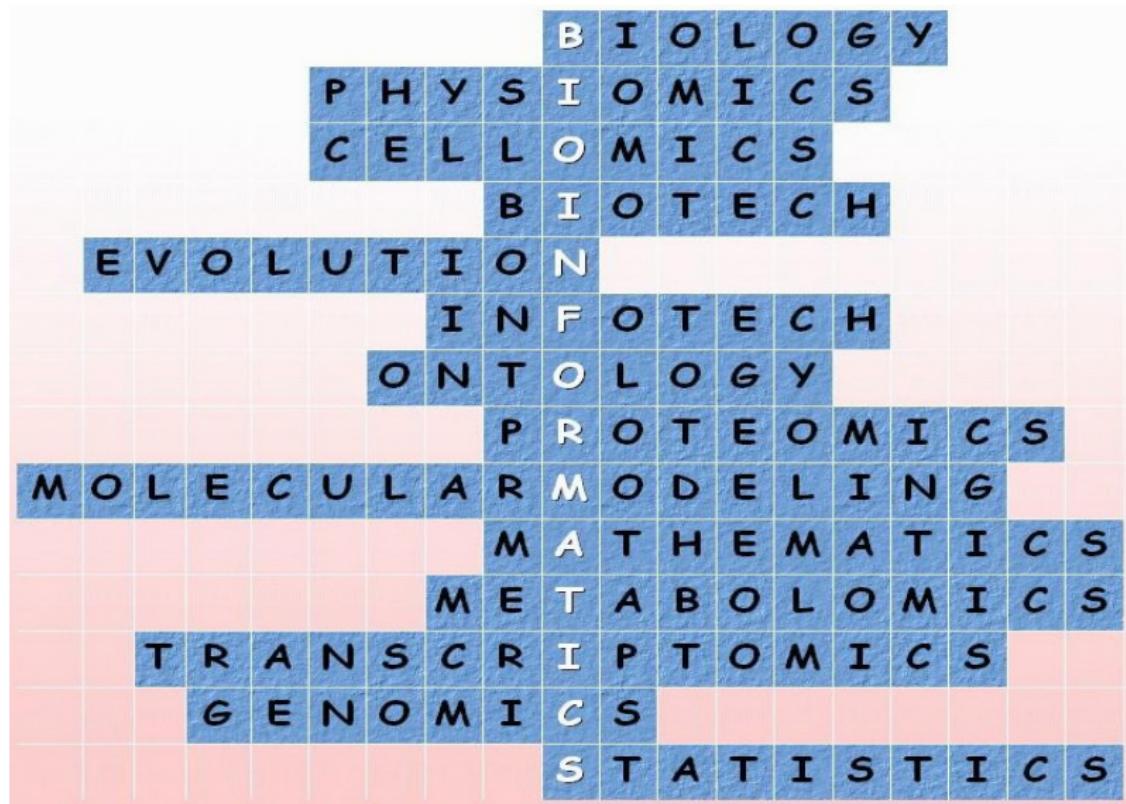
生物信息学可以定义为对分子生物学中两类信息流的研究：

第一类信息流源于分子生物学的中心法则：DNA 序列被转录为 mRNA 序列，后者被翻译为蛋白质序列。蛋白质序列继而折叠为具有功能的三维结构。按照达尔文演化理论，这些功能被生物体的环境所选择，从而驱动群体中 DNA 序列的进化。因此，第一类的生物信息学应用关注于中心法则中任一阶段的信息传递，包括 DNA 序列中基因的组织与控制、确定 DNA 中的转录单位、从序列预测蛋白质结构以及分子功能分析。

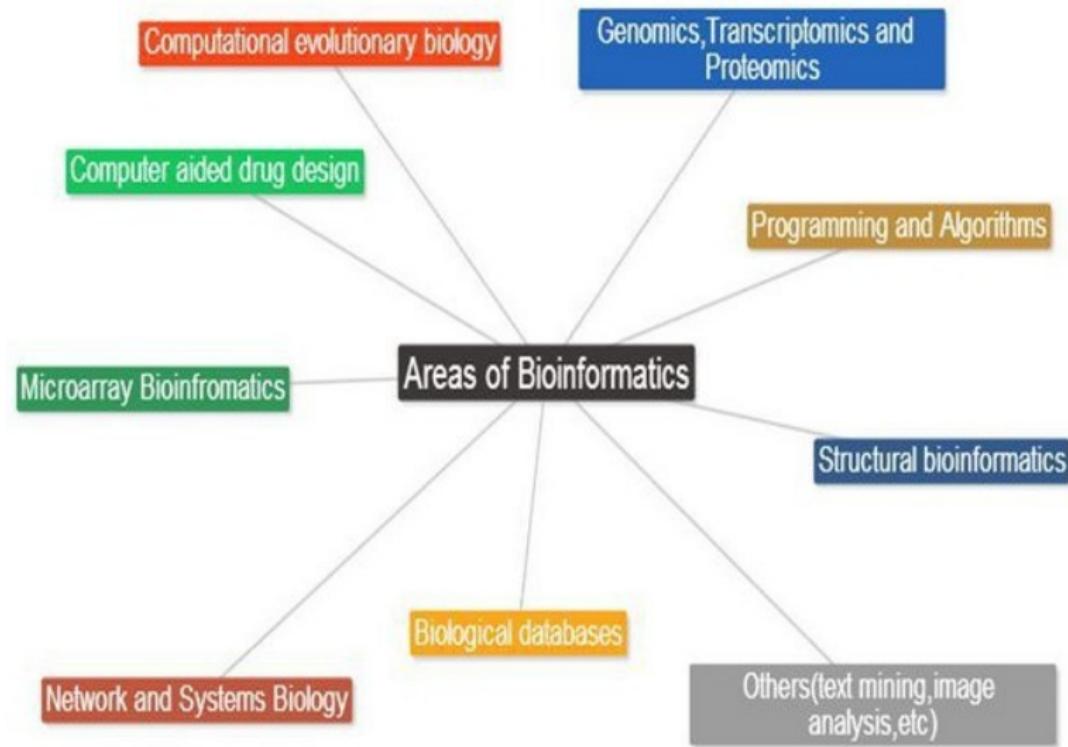
第二类信息流是基于科学方法：提出关于生物学活动的假设，设计实验以验证这些假设，评估结果与假设的相容性，然后根据实验数据对原假设作扩展或修正。第二类的生物信息学应用关注于这一流程中的信息传递，包括产生假设、设计实验、通过数据库将实验结果组织起来、检验数据与模型的相容性以及修正假设的各个系统。



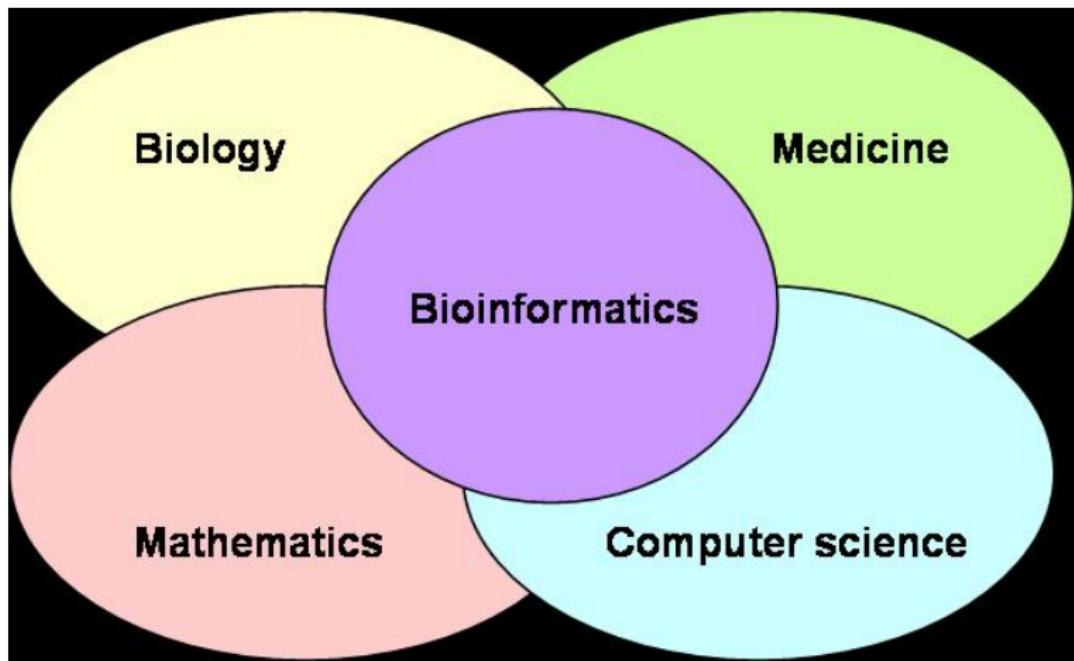
生物信息学 | 是什么？| 学科角度



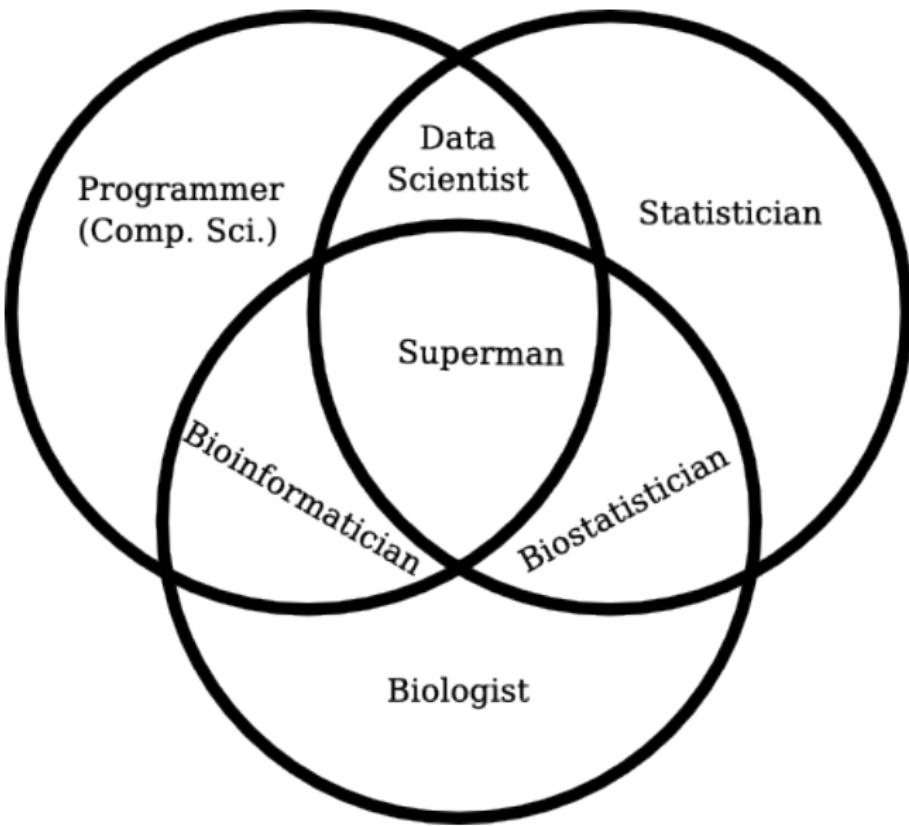
生物信息学 | 是什么？| 学科角度



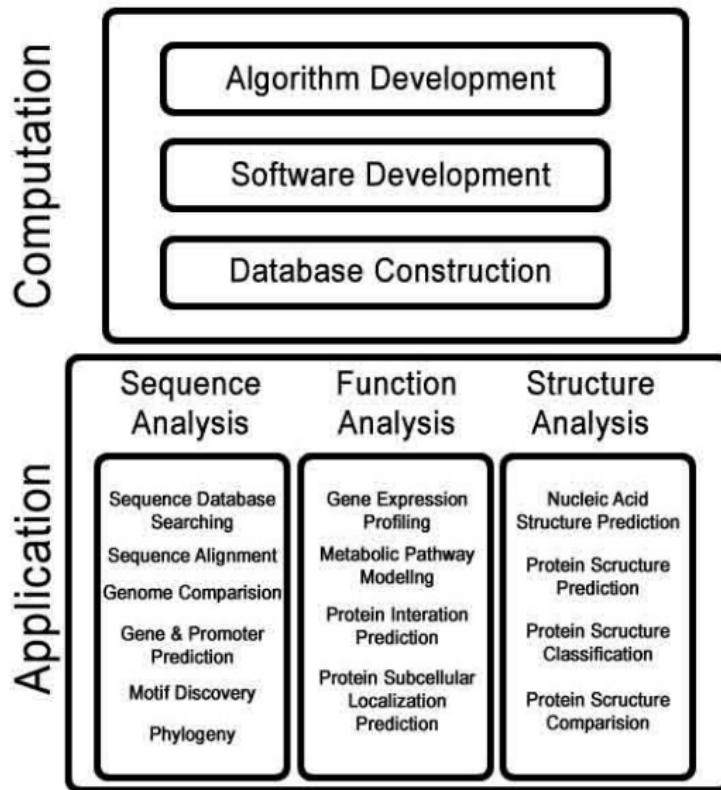
生物信息学 | 是什么？| 学科角度



生物信息学 | 是什么？| 技术角度



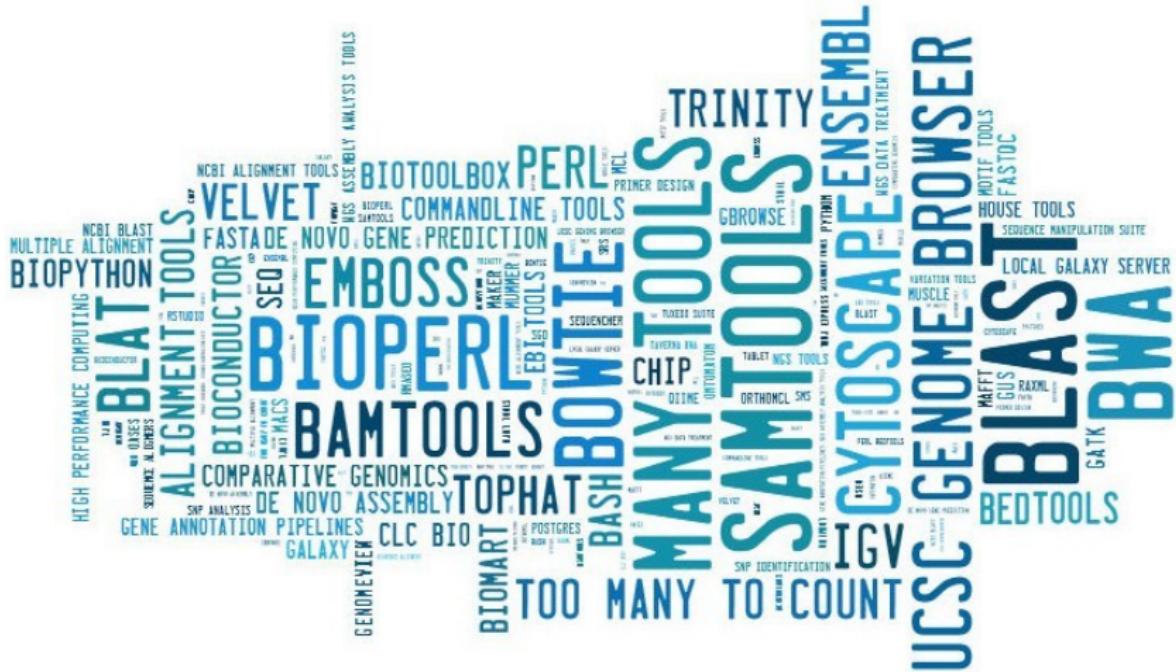
生物信息学 | 做什么？



生物信息学 | 资源 (数据库与工具)



生物信息学 | 资源 (技术与工具)



Cancer informatics Gene regulation
Personalized medicine Protein modeling
Computational biology Gene expression analysis
Image analysis Genomics and proteomics
Comparative genomics Gene expression databases
Epidemic models Computational drug discovery

Bioinformatics

Sequence analysis Bio-ontologies and semantics
Evolution and phylogenetics Structure prediction
Cheminformatics Next generation sequencing
Computational intelligence Transcriptomics
Biomedical engineering Amino acid sequencing
Structural bioinformatics Medical informatics
Microarrays
Visualization





生物信息学家 = 生命科学中的黑客！



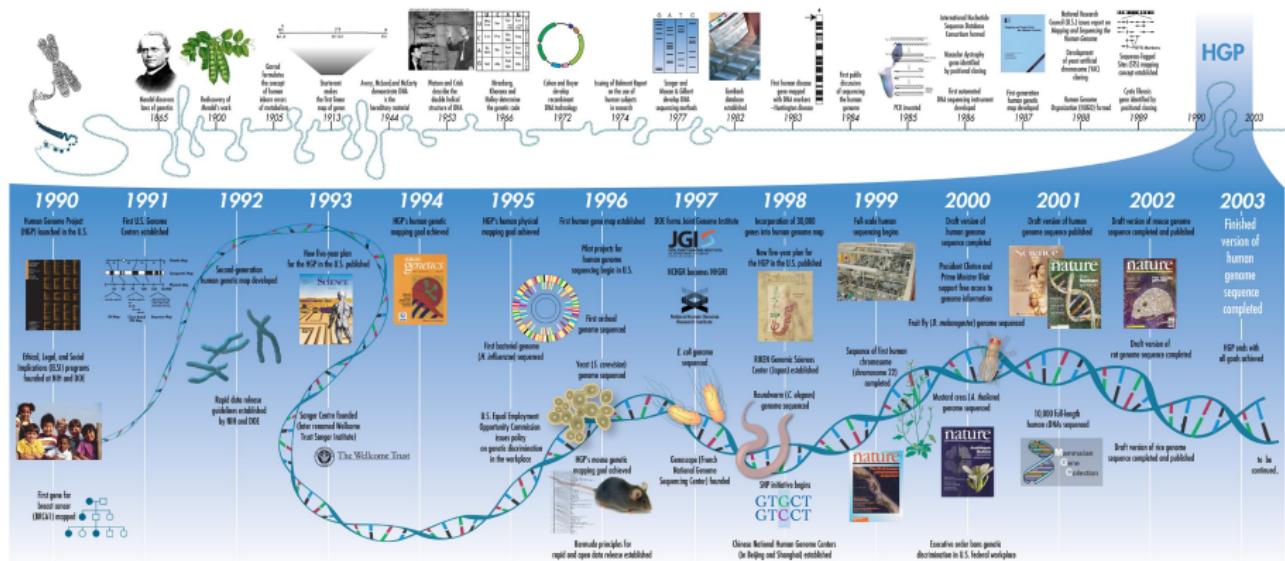
教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



生物学 | 历史



Central Dogma: DNA → RNA → Protein



DNA

transcription

CCTGAGCCAACCTATTGATGAA

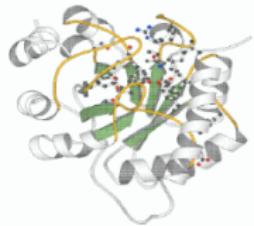
RNA

CCUGAGCCAACUAUUGAUUGAA

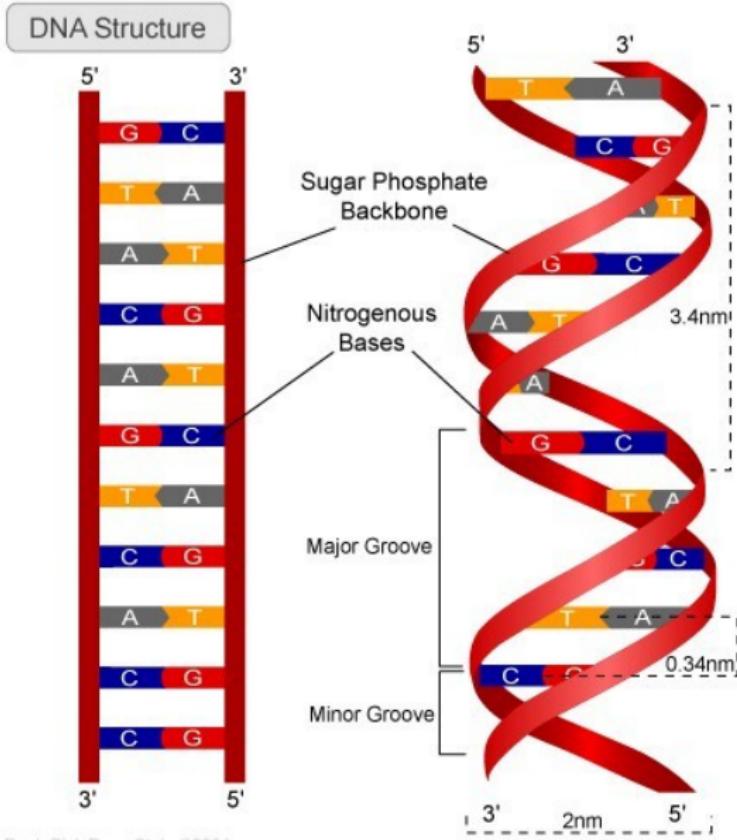
translation

Protein

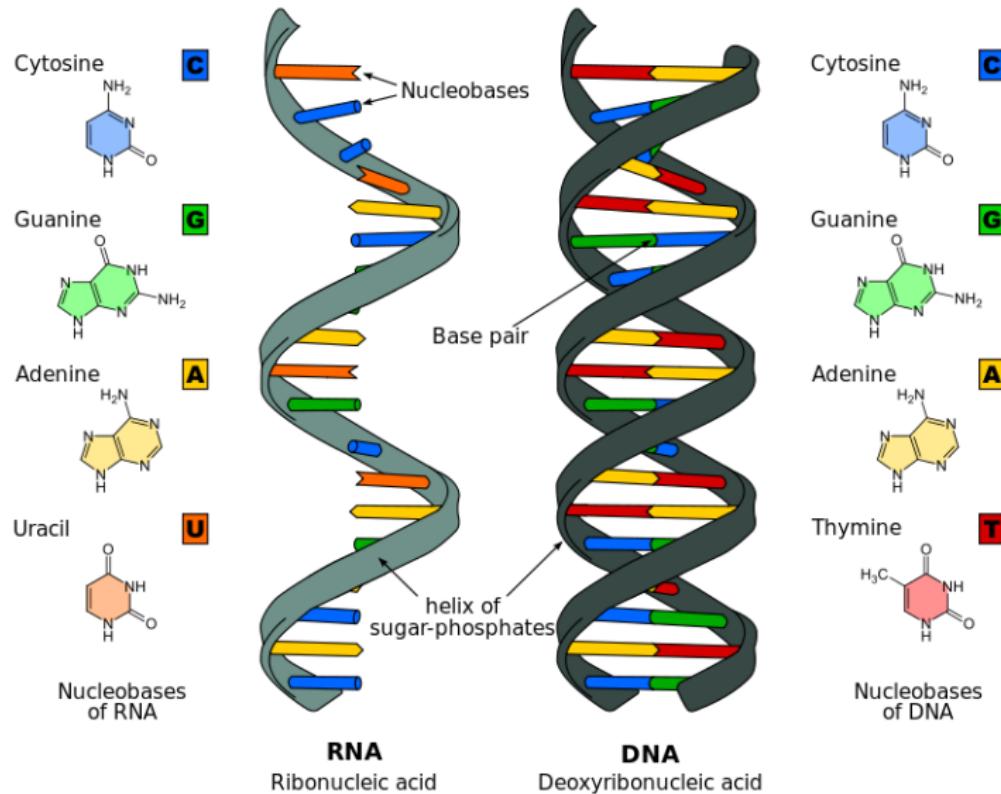
PEPTIDE



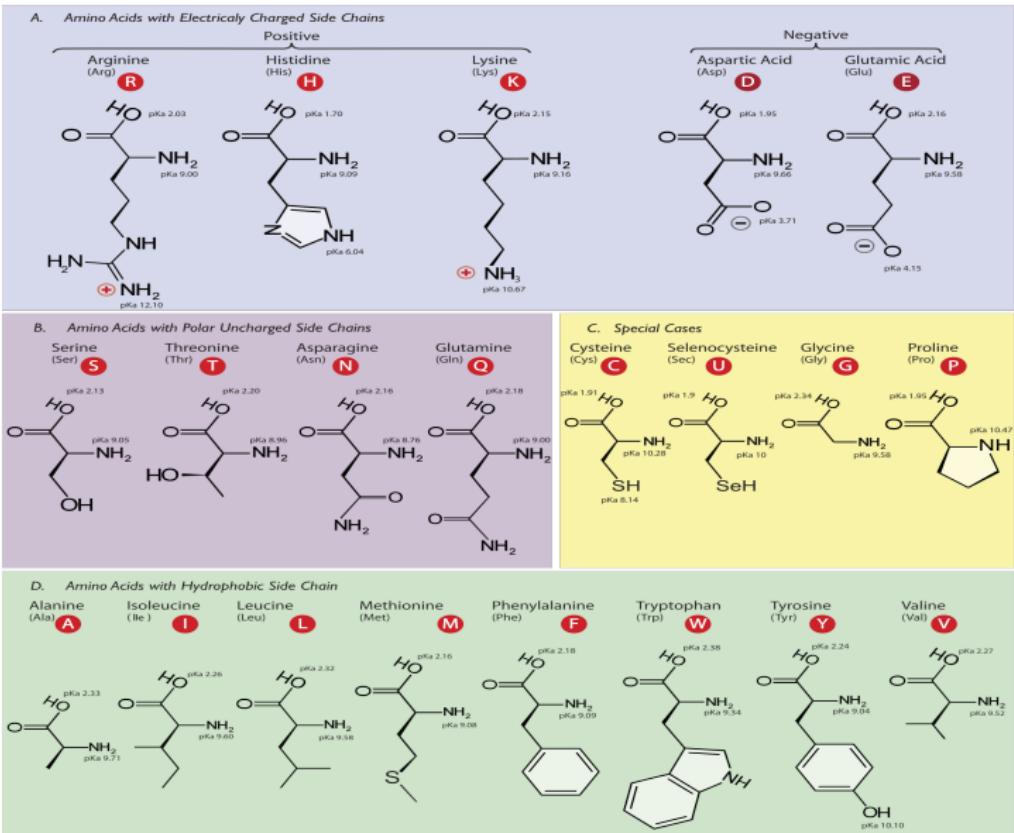
生物学 | DNA



生物学 | RNA



生物学 | 蛋白质 | 氨基酸



必需氨基酸

甲携来一本亮色书（甲硫氨酸、缬氨酸、赖氨酸、异亮氨酸、苯丙氨酸、亮氨酸、色氨酸、苏氨酸）

20 种氨基酸

苏缬亮异亮，苯丙属芳香。

还有色赖蛋，缺一人遭殃。

(以上是必需氨基酸)

丙组丝甘半，天谷建酸胺。

精酪加一脯，二十氨基酸。

(以上是非必需氨基酸)



必需氨基酸

甲携来一本亮色书（甲硫氨酸、缬氨酸、赖氨酸、异亮氨酸、苯丙氨酸、亮氨酸、色氨酸、苏氨酸）

20 种氨基酸

苏缬亮异亮，苯丙属芳香。

还有色赖蛋，缺一人遭殃。

(以上是必需氨基酸)

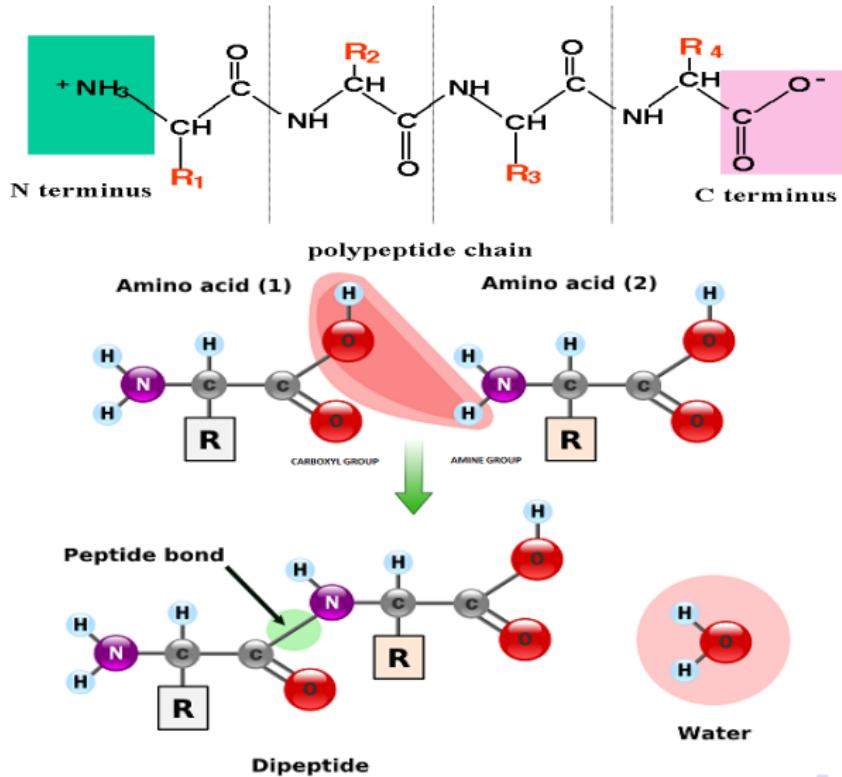
丙组丝甘半，天谷建酸胺。

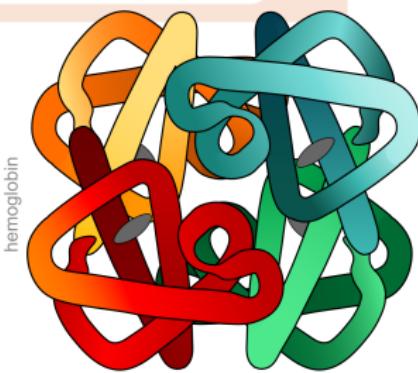
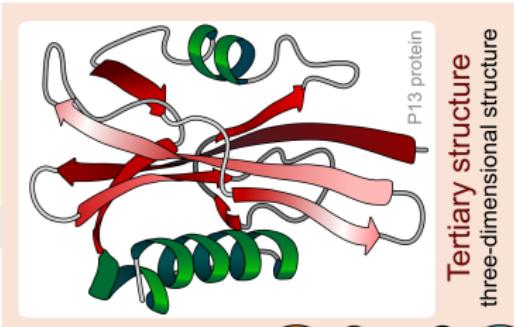
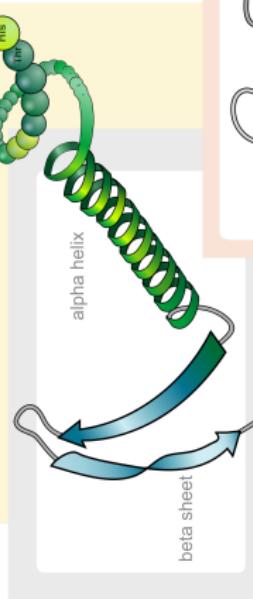
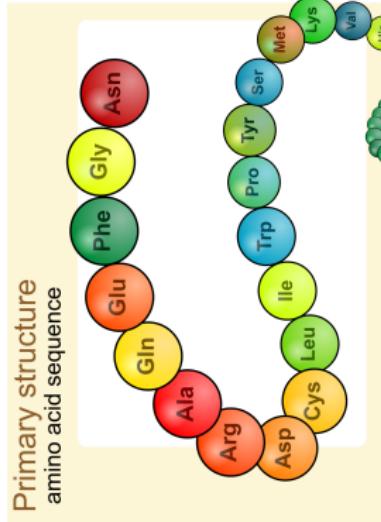
精酪加一脯，二十氨基酸。

(以上是非必需氨基酸)



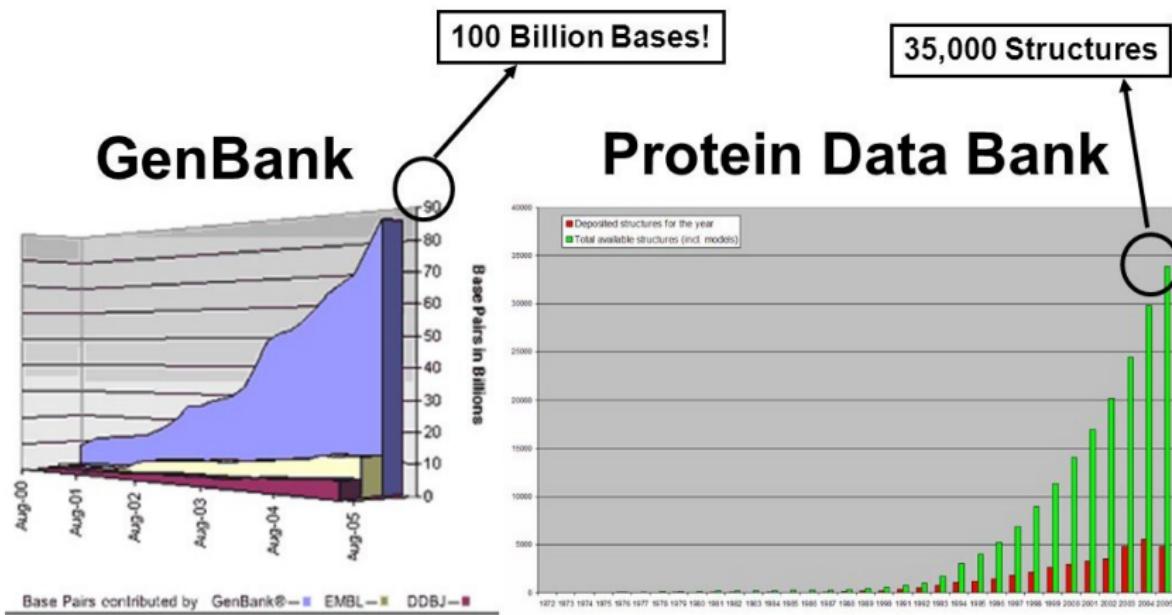
Peptide = chain of amino acids





Quaternary structure
complex of protein molecules



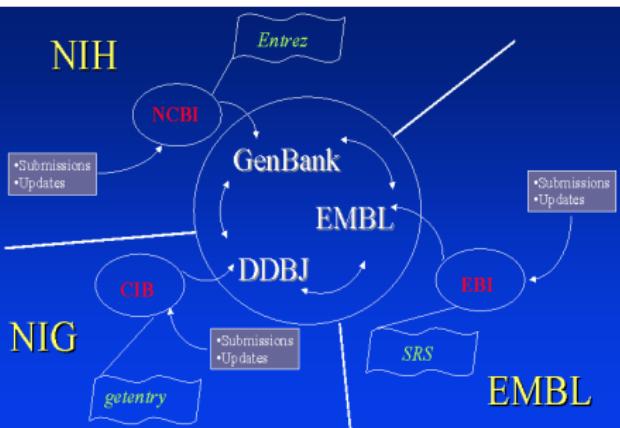
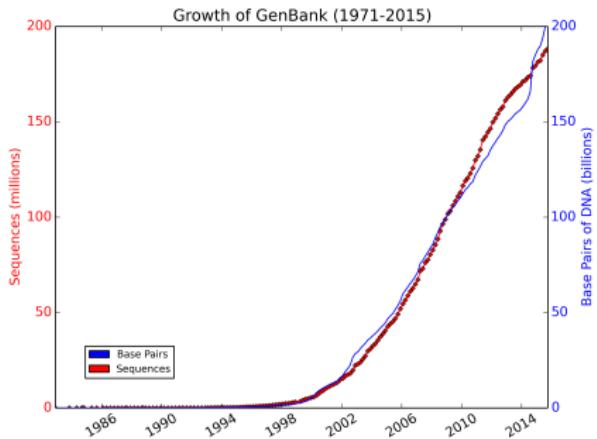


www.ncbi.nlm.nih.gov/Genbank

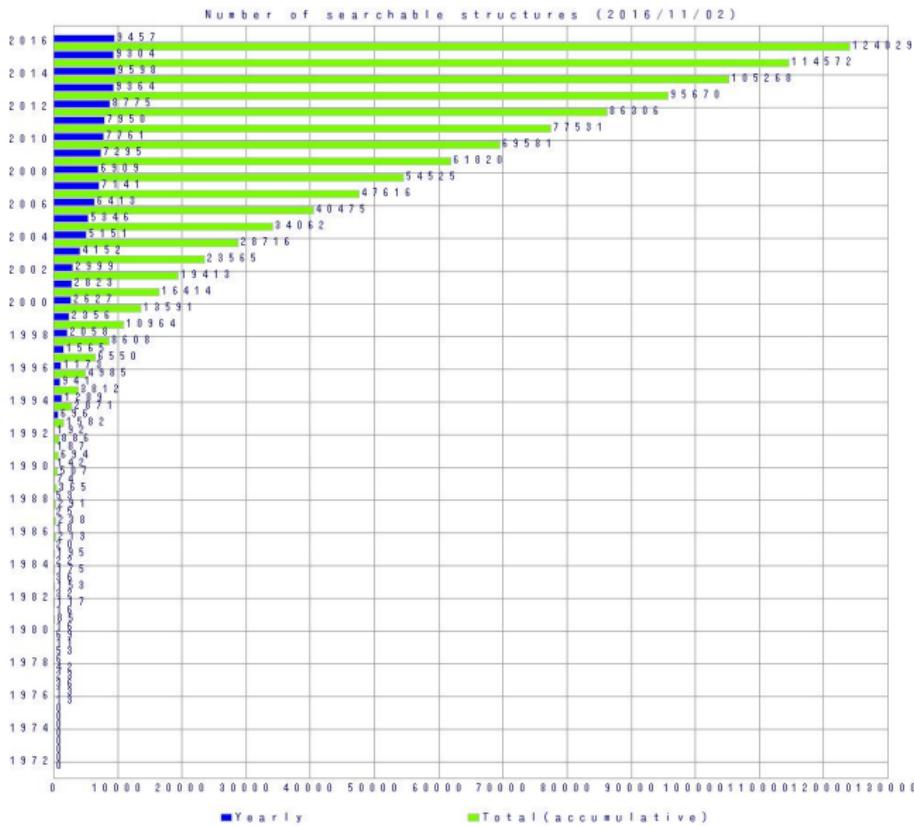
www.rcsb.org/pdb/holdings.html



生物学 | 数据库 | GenBank



生物学 | 数据库 | PDB



生物学 | 数据格式

Historically, bioinformatics has always used text files to store data.



HMM profile

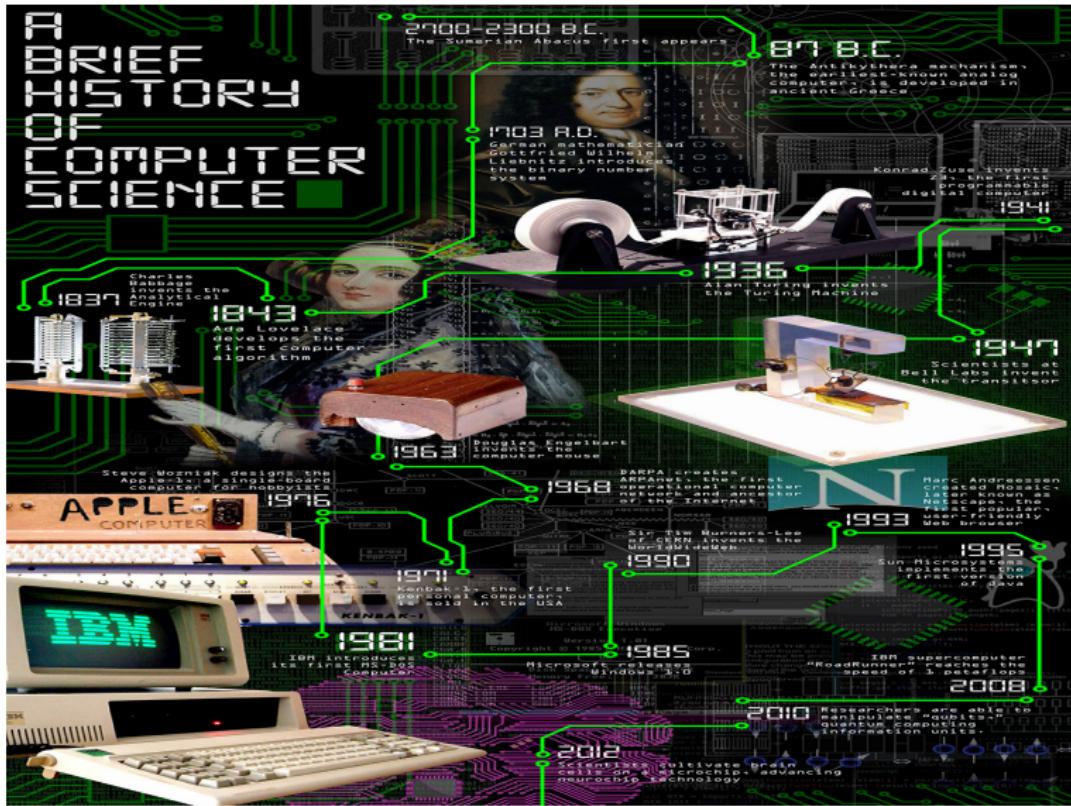


教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题





in vivo

In vivo 为拉丁文 “在活体内” 之意。在科学文献中，*in vivo* 常指进行于完整且存活的个体内的组织的实验，以区别在生物体上移除下来的组织或死亡的组织上进行的实验（对应的拉丁文为 *in vitro*）。

in vitro

In vitro 是拉丁语中 “在玻璃里” 的意思，意指进行或发生于试管内的实验与实验技术。更广义的意思，则指活生物体之外的环境中的操作。

in silico

In silico 是指 “在硅之中”，也就是说 “进行于电脑中，或是经由电脑模拟” 之意，此用语是衍生自另外两个在生物学上常用的短语：*in vivo*（生物活体内）及 *in vitro*（生物活体外）。

in vivo

In vivo 为拉丁文 “在活体内” 之意。在科学文献中，*in vivo* 常指进行于完整且存活的个体内的组织的实验，以区别在生物体上移除下来的组织或死亡的组织上进行的实验（对应的拉丁文为 *in vitro*）。

in vitro

In vitro 是拉丁语中 “在玻璃里” 的意思，意指进行或发生于试管内的实验与实验技术。更广义的意思，则指活生物体之外的环境中的操作。

in silico

In silico 是指 “在硅之中”，也就是说 “进行于电脑中，或是经由电脑模拟” 之意，此用语是衍生自另外两个在生物学上常用的短语：*in vivo*（生物活体内）及 *in vitro*（生物活体外）。



in vivo

In vivo 为拉丁文 “在活体内” 之意。在科学文献中，*in vivo* 常指进行于完整且存活的个体内的组织的实验，以区别在生物体上移除下来的组织或死亡的组织上进行的实验（对应的拉丁文为 *in vitro*）。

in vitro

In vitro 是拉丁语中 “在玻璃里” 的意思，意指进行或发生于试管内的实验与实验技术。更广义的意思，则指活生物体之外的环境中的操作。

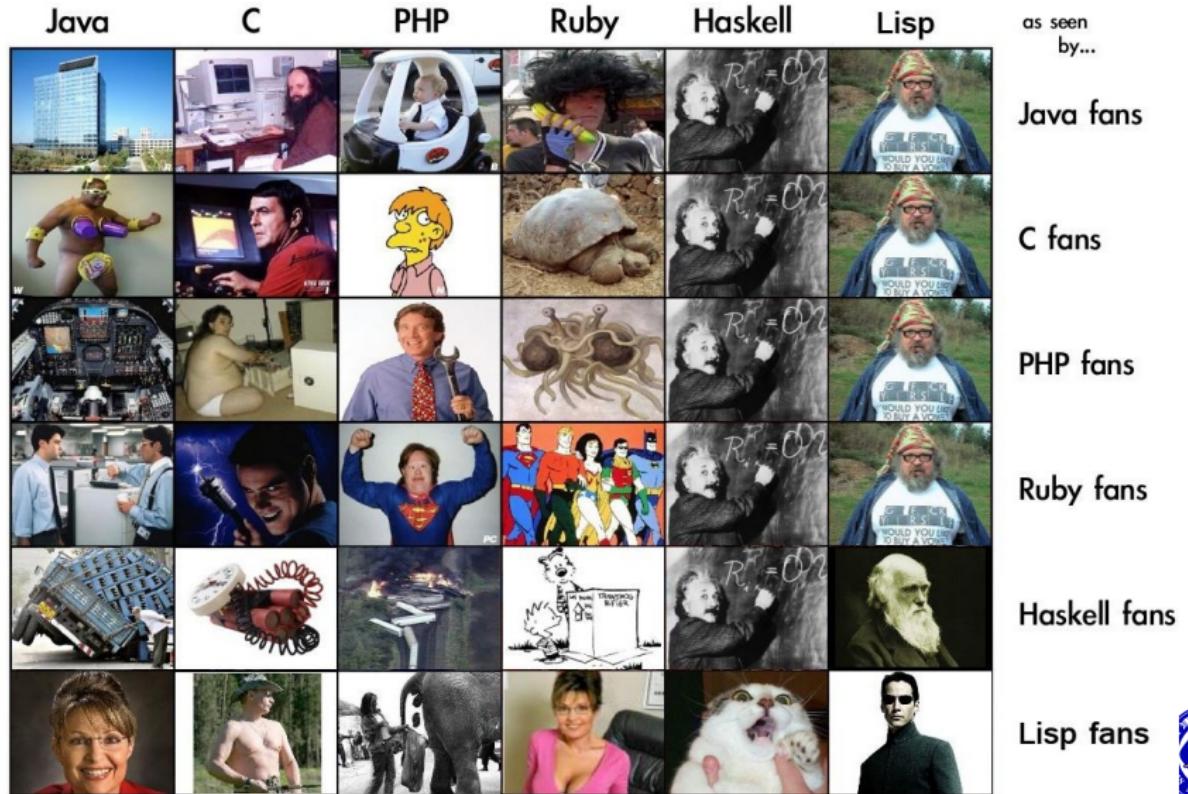
in silico

In silico 是指 “在硅之中”，也就是说 “进行于电脑中，或是经由电脑模拟” 之意，此用语是衍生自另外两个在生物学上常用的短语：*in vivo*（生物活体内）及 *in vitro*（生物活体外）。





计算机科学 | 编程语言 | 崇拜/鄙视链



计算机科学 | 编程语言 | 特色

	C++		JavaScript
	Java/C#		PHP(Without MySQL)
	Ruby		Pascal
	Perl		Lisp
	Visual Basic		Haskell
	Python		C

语言	宗教	超级英雄	《哈利波特》
C	犹太教	—	伏地魔
C++	伊斯兰教	—	西弗勒斯·斯内普
Java	正统基督教	万磁王	洛雷斯·乌姆里奇
Lisp	佛教	Xavier 教授	—
Perl	巫毒教	—	罗恩·韦斯莱
PHP	Cafeteria 基督教	小丑王	德拉科·马尔福
Python	人文主义	蝙蝠侠	哈利·波特
Ruby	新异教主义	钢铁侠	—
shell	—	—	鲁伯·海格

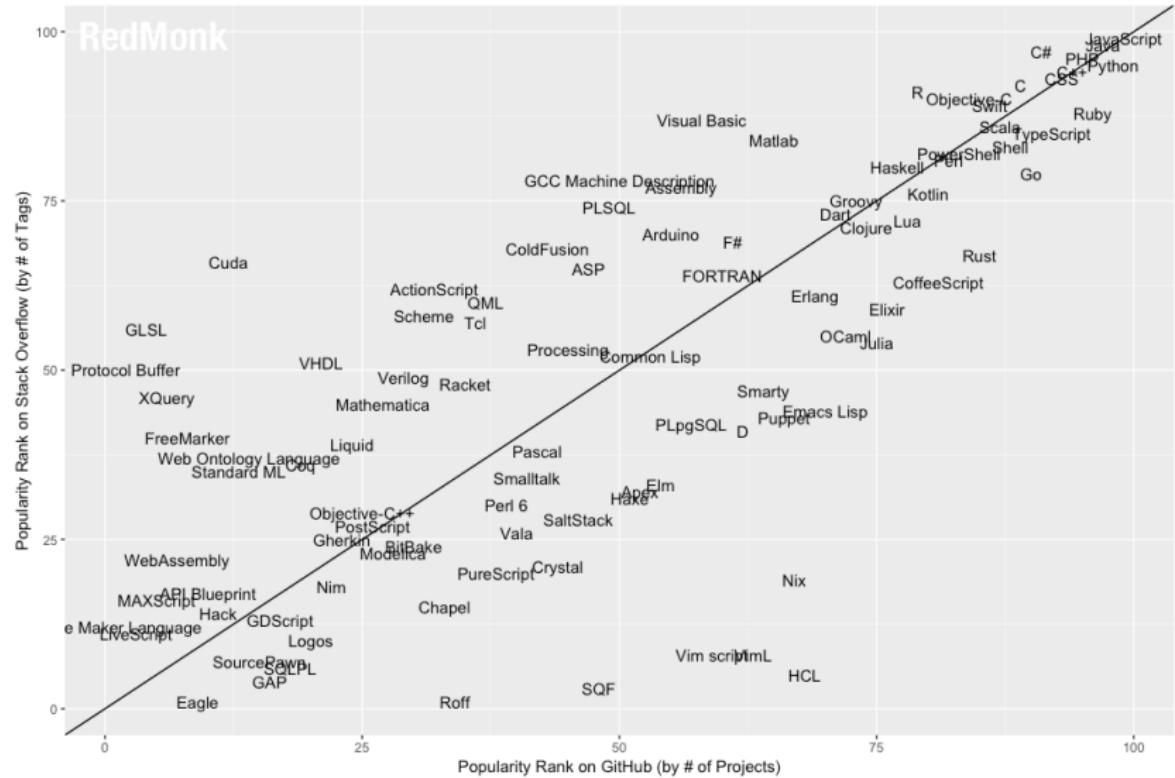


计算机科学 | 编程语言 | 如果是…… (续)

语言	女人	武器	船
C	霸道女总裁	M1 式加兰德步枪	核潜艇
C++	—	双截棍	—
Java	娇妻贤内助	M240 通用弹夹式自动机枪	大货船
Lisp	女博士	剃须刀	—
Perl	—	燃烧弹	拖船
PHP	—	水管子	竹筏
Python	万人迷	双管枪	—
Ruby	—	宝刀	摩托艇
shell	女公务员	锤子	—



RedMonk Q319 Programming Language Rankings



计算机科学 | 编程语言 | 排名 | 2019

Aug 2019	Aug 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.028%	-0.85%
2	2		C	15.154%	+0.19%
3	4	▲	Python	10.020%	+3.03%
4	3	▼	C++	6.057%	-1.41%
5	6	▲	C#	3.842%	+0.30%
6	5	▼	Visual Basic .NET	3.695%	-1.07%
7	8	▲	JavaScript	2.258%	-0.15%
8	7	▼	PHP	2.075%	-0.85%
9	14	▲	Objective-C	1.690%	+0.33%
10	9	▼	SQL	1.625%	-0.69%
11	15	▲	Ruby	1.316%	+0.13%
12	13	▲	MATLAB	1.274%	-0.09%
13	44	▲	Groovy	1.225%	+1.04%
14	12	▼	Delphi/Object Pascal	1.194%	-0.18%
15	10	▼	Assembly language	1.114%	-0.30%
16	19	▲	Visual Basic	1.025%	+0.10%
17	17		Go	0.973%	-0.02%
18	11	▼	Swift	0.890%	-0.49%
19	16	▼	Perl	0.860%	-0.31%
20	18	▼	R	0.822%	-0.14%



计算机科学 | 编程语言 | 排名 | 历史

Programming Language	2019	2014	2009	2004	1999	1994	1989
Java	1	2	1	1	14	-	-
C	2	1	2	2	1	1	1
Python	3	7	5	7	24	21	-
C++	4	4	3	3	2	2	2
Visual Basic .NET	5	9	-	-	-	-	-
C#	6	5	6	6	19	-	-
JavaScript	7	8	8	8	16	-	-
PHP	8	6	4	5	-	-	-
SQL	9	-	-	89	-	-	-
Objective-C	10	3	31	38	-	-	-
Perl	16	11	7	4	3	10	22
Lisp	32	13	19	13	12	5	3
Pascal	220	16	14	88	6	3	20



How Perl Saved the Human Genome Project

by Lincoln Stein

Reprinted courtesy of the Perl Journal, <http://www.tpj.com>

Lincoln Stein's website is <http://stein.cshl.org>

DATE: Early February, 1996

LOCATION: Cambridge, England, in the conference room of the largest DNA sequencing center in Europe.

OCCASION: A high level meeting between the computer scientists of this center and the largest DNA sequencing center in the United States.

THE PROBLEM: Although the two centers use almost identical laboratory techniques, almost identical databases, and almost identical data analysis tools, they still can't interchange data or meaningfully compare results.

THE SOLUTION: Perl.

How Perl Saved the Human Genome Project

Perl and the human genome are almost perfectly matched; both are almost incomprehensible, with no central design, accreted haphazardly over a long time.



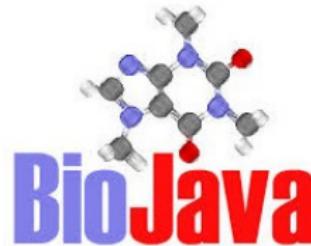
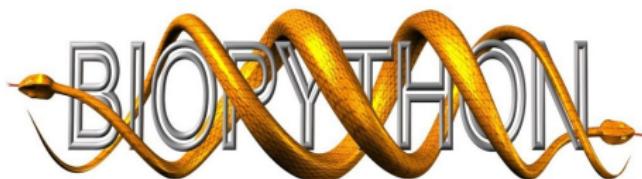
生物信息学常用编程语言

Perl (1987) 、 Python (1991) 、 Ruby (1995) 、 Java (1995)



生物信息学专用

BioPerl、Biopython、BioRuby、BioJava



教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



不需要！

- 有现成的工具可以使用
- 五湖四海皆兄弟
- 有钱能使磨推鬼
-

需要！

- 没有现成的工具
- 兄弟们都在忙着混江湖
- 工资涨如龟速，物价飙如赤兔
-



不需要！

- 有现成的工具可以使用
- 五湖四海皆兄弟
- 有钱能使磨推鬼
-

需要！

- 没有现成的工具
- 兄弟们都在忙着混江湖
- 工资涨如龟速，物价飙如赤兔
-



- 有助于理解现有工具的配置与个性化修改
- 编写程序来批量运行现有程序
- 数据分析很简单，前期数据处理很难很难……（二八定律，80/20 法则，帕雷托法则）
- 增强研究工作的可重复性
- 数据越来越大，时间越来越少
- 人生苦短，学习编程
-



编程有助于逻辑思维的锻炼，系统观的形成，以及创造能力和解决问题能力的培养。但这些都是潜移默化的，需要有一个积累的过程。

- 试错力：编程的试错成本比较低，在一次次修复 bug 的过程中，慢慢地就不怕去试错了
- 创造力：编程可以把想象变为现实，不断去创造奇迹
- 学习能力：编程独特的操作和思考性，不仅会让编程者爱上学习，还会帮助养成严谨、认真的学习习惯
- 逻辑思维：一个人语言组织、领导能力的基础
- 科学思维：编程就是用科学的眼光去思考问题、分析问题，更全面地去看待这个世界
-



编程思维

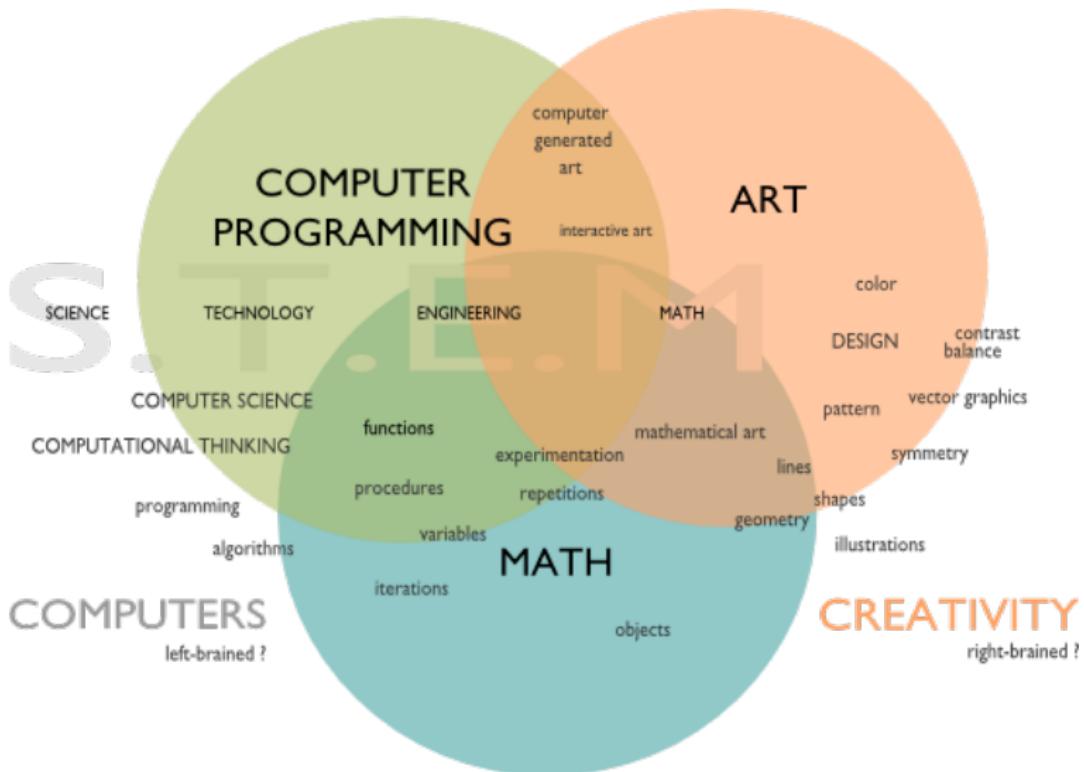
编程思维 (computational thinking) 就是 “理解问题-找出路径” 的思维过程，它由分解、模式识别、抽象、算法四个步骤组成。

通过这四个步骤，一个棘手的复杂问题先被拆解成一系列好解决的小问题；每一个小问题被单独检视、思考，搜索解决方法；然后，聚焦几个重要节点，忽视小细节，形成解决思路；最后，设计步骤，执行——问题解决！

通过编程做一个作品的过程，就是自己创造一个事物的过程。最起码首先要在心里大致构造出自己想要的作品模样，然后开始思考第一步该怎么做，第二步该怎么做。



编程 | 与艺术



每一位程序员都是一位艺术家 每一个程序都是一件艺术品 编写程序 = 艺术创作！



If you program enough,
it can change the way you look at the world ...



任务

- ① 同时精确测量粒子的位置和动量
- ② 寻找第四种可以镶嵌平面的凸六边形
- ③ 寻找可以镶嵌平面的边数大于 6 的凸多边形
- ④ 分析敲除所有必需基因后的基因表达

原理

- ① 海森堡不确定性原理
- ② 有且只有三种凸六边形可以镶嵌平面
- ③ 边数大于 6 的凸多边形都无法镶嵌平面
- ④ 都说了是必需基因……



任务

- ① 同时精确测量粒子的位置和动量
- ② 寻找第四种可以镶嵌平面的凸六边形
- ③ 寻找可以镶嵌平面的边数大于 6 的凸多边形
- ④ 分析敲除所有必需基因后的基因表达

原理

- ① 海森堡不确定性原理
- ② 有且只有三种凸六边形可以镶嵌平面
- ③ 边数大于 6 的凸多边形都无法镶嵌平面
- ④ 都说了是必需基因……



任务

- ① 同时精确测量粒子的位置和动量
- ② 寻找第四种可以镶嵌平面的凸六边形
- ③ 寻找可以镶嵌平面的边数大于 6 的凸多边形
- ④ 分析敲除所有必需基因后的基因表达

原理

- ① 海森堡不确定性原理
- ② 有且只有三种凸六边形可以镶嵌平面
- ③ 边数大于 6 的凸多边形都无法镶嵌平面
- ④ 都说了是必需基因……



任务

- ① 同时精确测量粒子的位置和动量
- ② 寻找第四种可以镶嵌平面的凸六边形
- ③ 寻找可以镶嵌平面的边数大于 6 的凸多边形
- ④ 分析敲除所有必需基因后的基因表达

原理

- ① 海森堡不确定性原理
- ② 有且只有三种凸六边形可以镶嵌平面
- ③ 边数大于 6 的凸多边形都无法镶嵌平面
- ④ 都说了是必需基因……



任务

- ① 破解 RSA 秘钥
- ② 暴力破解由 94 个字符（26 小写、26 大写、10 数字、32 标点）随机组合成的长 12 个字符的密码
- ③ 蛋白质折叠中通过随机尝试找到总能量最低的构象状态

原因

- ① 对极大整数做因数分解非常困难（破解 RSA-2048（2048-bit）的密钥可能需要耗费传统电脑 10 亿年的时间，而量子计算机只需要 100 秒就可以完成。）【量子计算机 + 秀尔算法】/【黎曼猜想】
- ② 普通台式机，每秒运算 40 亿次，需要大约 30 万年以上才能破解
- ③ 利文索尔佯谬（Levinthal's paradox）：100 氨基酸，每个 2 种构象，每次尝试耗时 10^{-13} s，穷举需要 40 亿年【深度学习？】

任务

- ① 破解 RSA 秘钥
- ② 暴力破解由 94 个字符（26 小写、26 大写、10 数字、32 标点）随机组合成的长 12 个字符的密码
- ③ 蛋白质折叠中通过随机尝试找到总能量最低的构象状态

原因

- ① 对极大整数做因数分解非常困难（破解 RSA-2048（2048-bit）的密钥可能需要耗费传统电脑 10 亿年的时间，而量子计算机只需要 100 秒就可以完成。）【量子计算机 + 秀尔算法】/【黎曼猜想】
- ② 普通台式机，每秒运算 40 亿次，需要大约 30 万年以上才能破解
- ③ 利文索尔佯谬（Levinthal's paradox）：100 氨基酸，每个 2 种构象，每次尝试耗时 10^{-13} s，穷举需要 40 亿年【深度学习？】

任务

- ① 破解 RSA 秘钥
- ② 暴力破解由 94 个字符（26 小写、26 大写、10 数字、32 标点）随机组合成的长 12 个字符的密码
- ③ 蛋白质折叠中通过随机尝试找到总能量最低的构象状态

原因

- ① 对极大整数做因数分解非常困难（破解 RSA-2048（2048-bit）的密钥可能需要耗费传统电脑 10 亿年的时间，而量子计算机只需要 100 秒就可以完成。）【量子计算机 + 秀尔算法】/【黎曼猜想】
- ② 普通台式机，每秒运算 40 亿次，需要大约 30 万年以上才能破解
- ③ 利文索尔佯谬（Levinthal's paradox）：100 氨基酸，每个 2 种构象，每次尝试耗时 10^{-13} s，穷举需要 40 亿年【深度学习？】

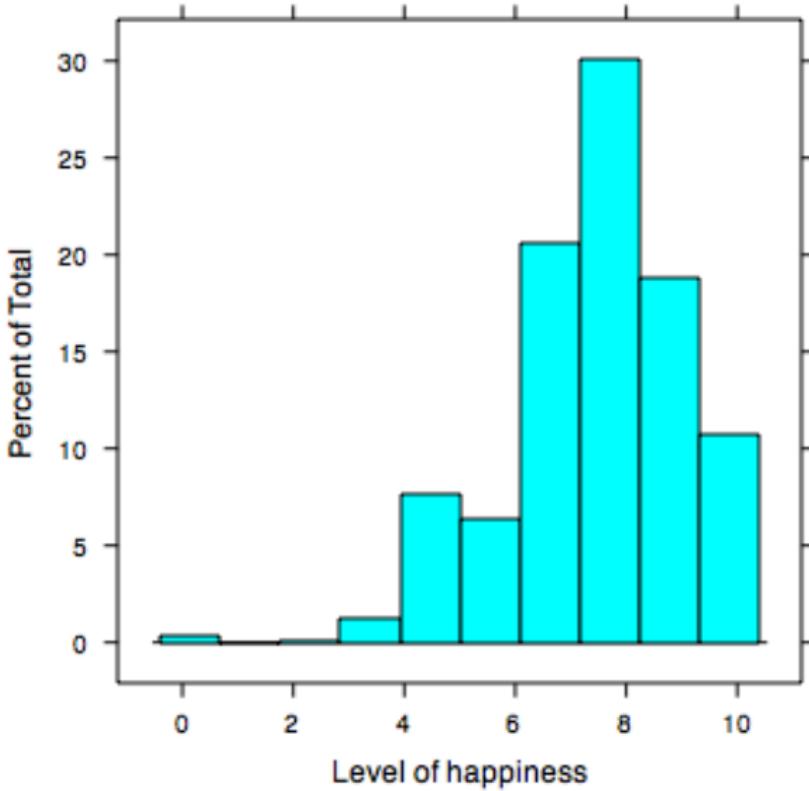
教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

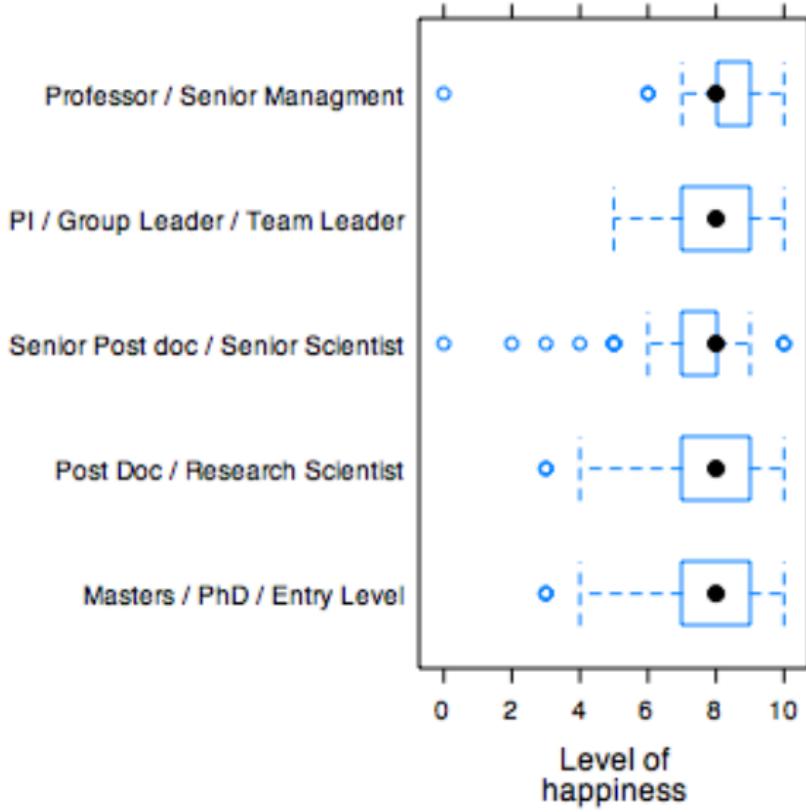
- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



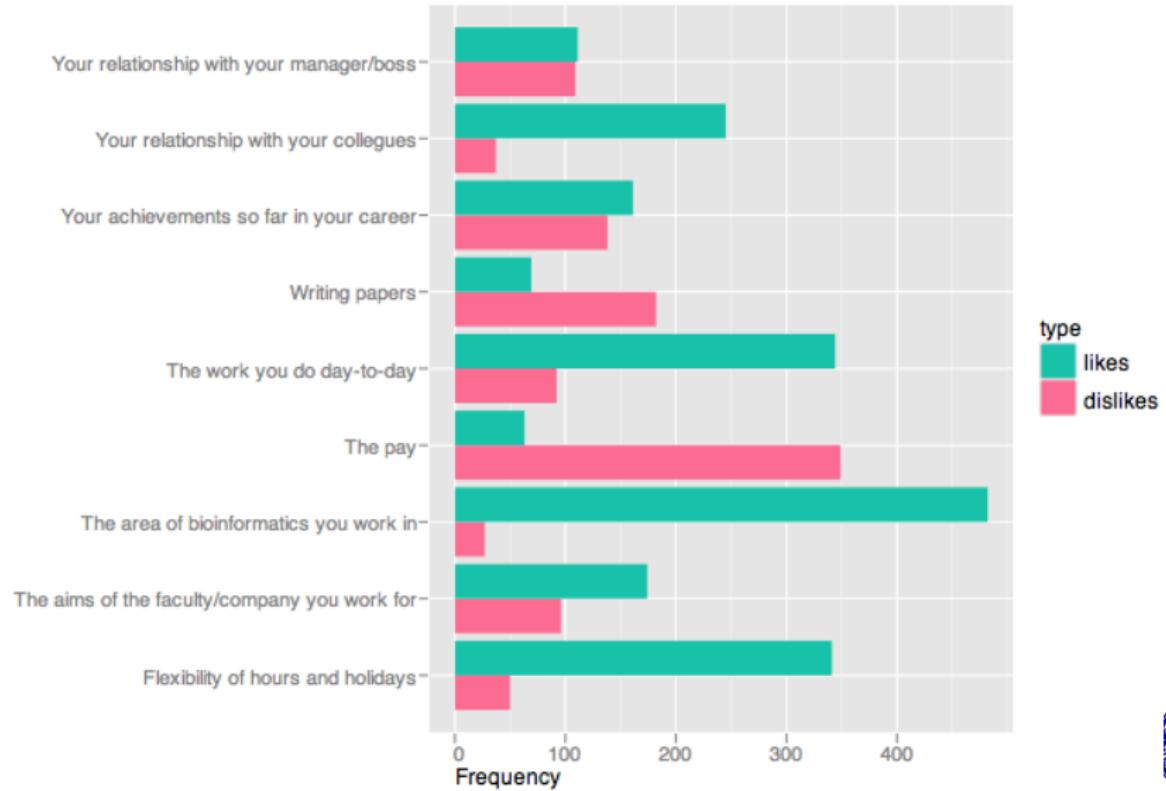
2008 | happiness

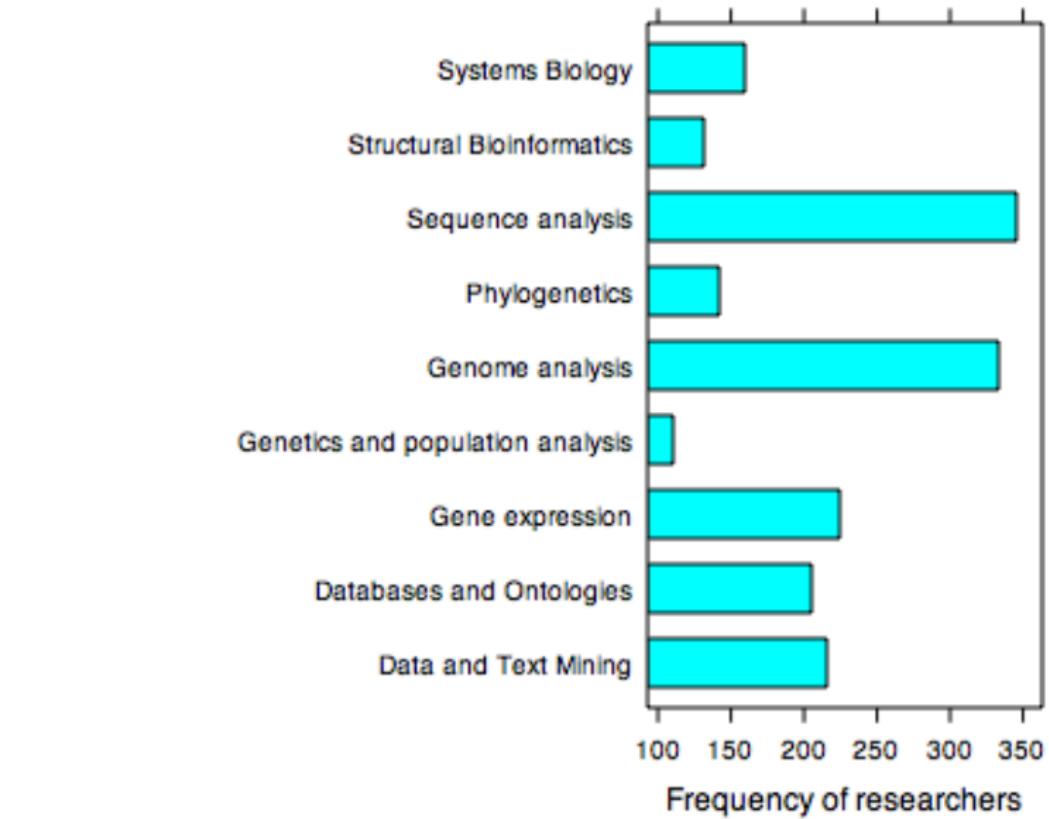


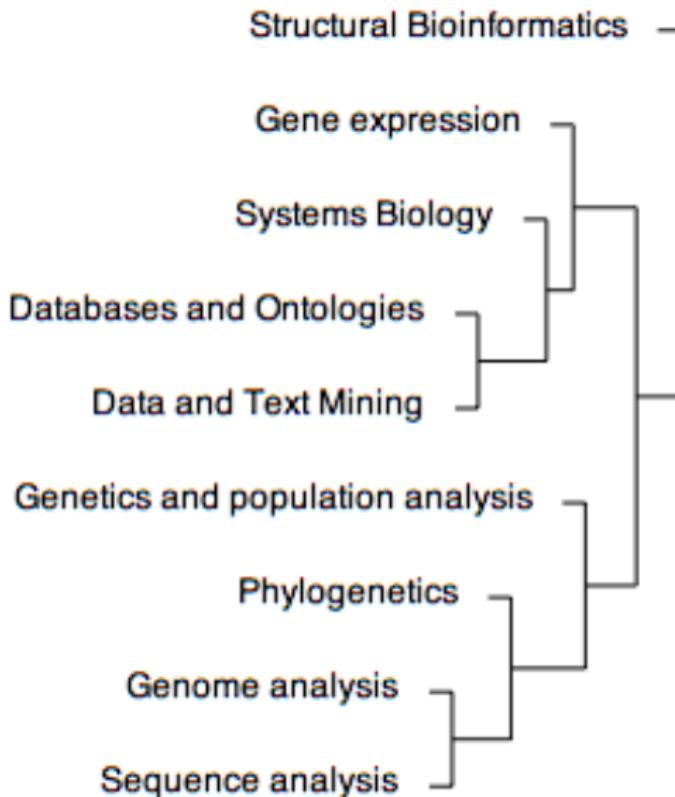
2008 | happiness vs. career position



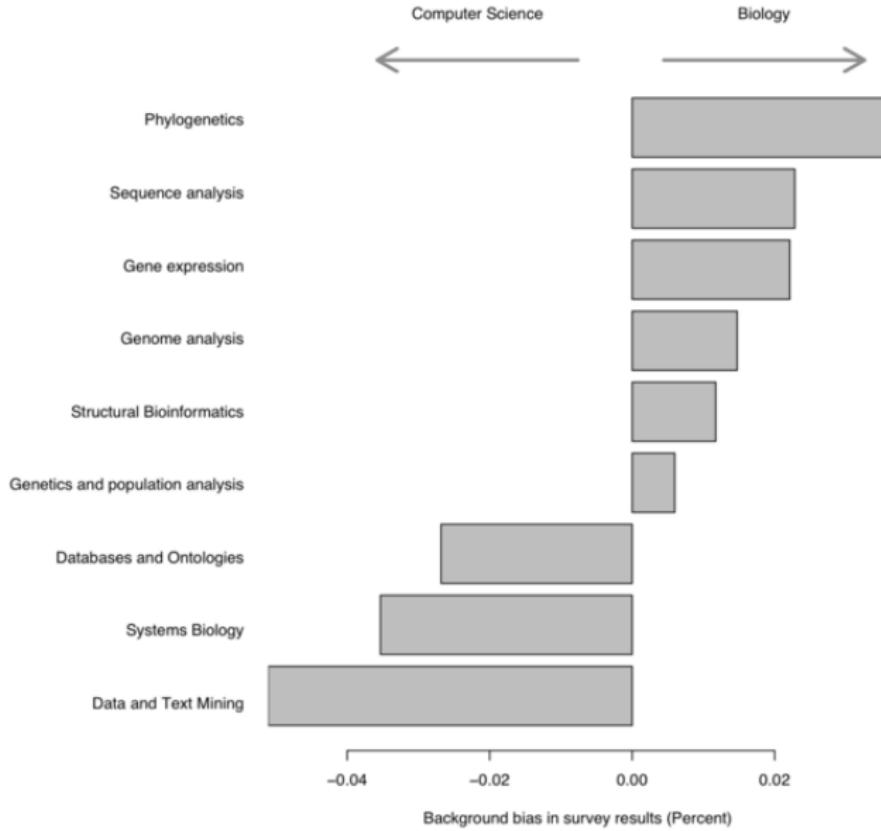
2008 | likes vs. dislikes



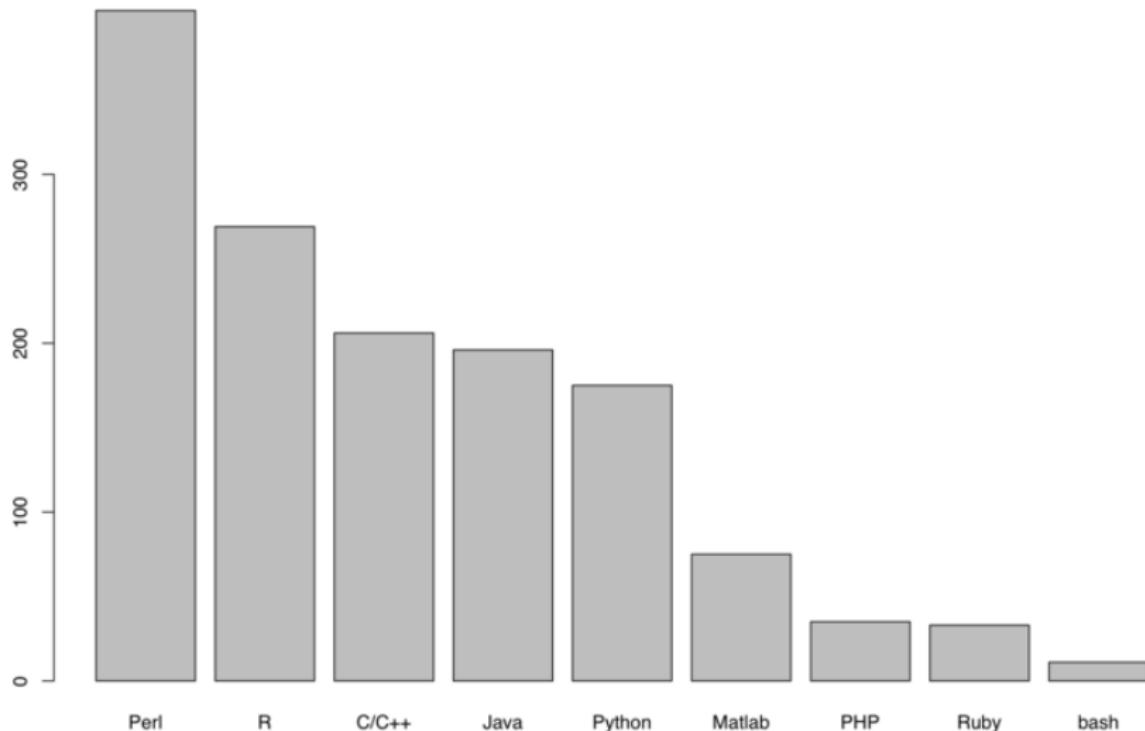




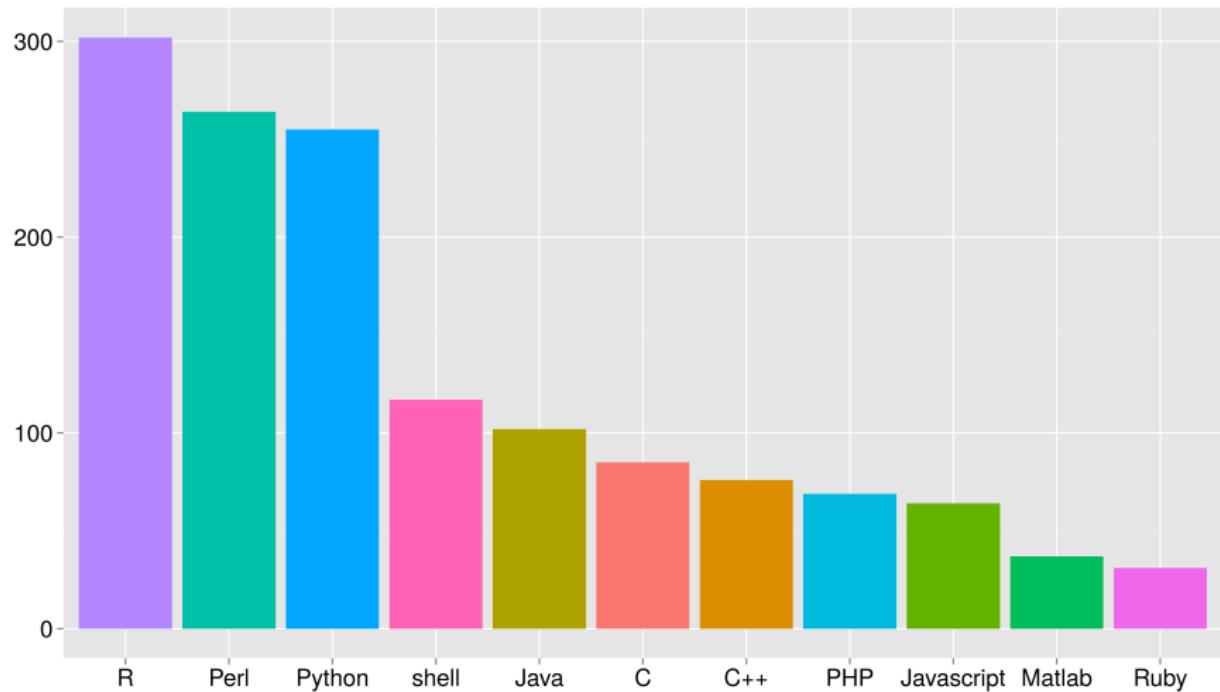
2008 | background

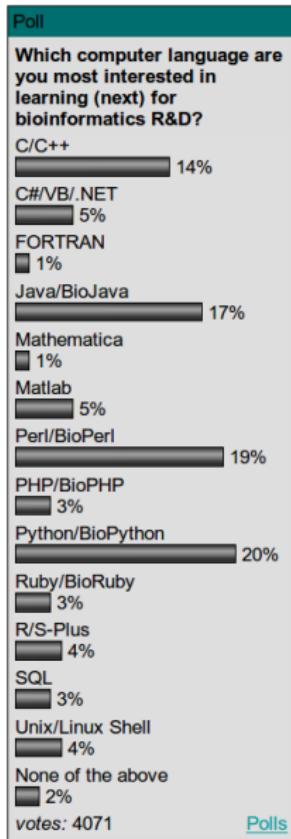


2008 | language

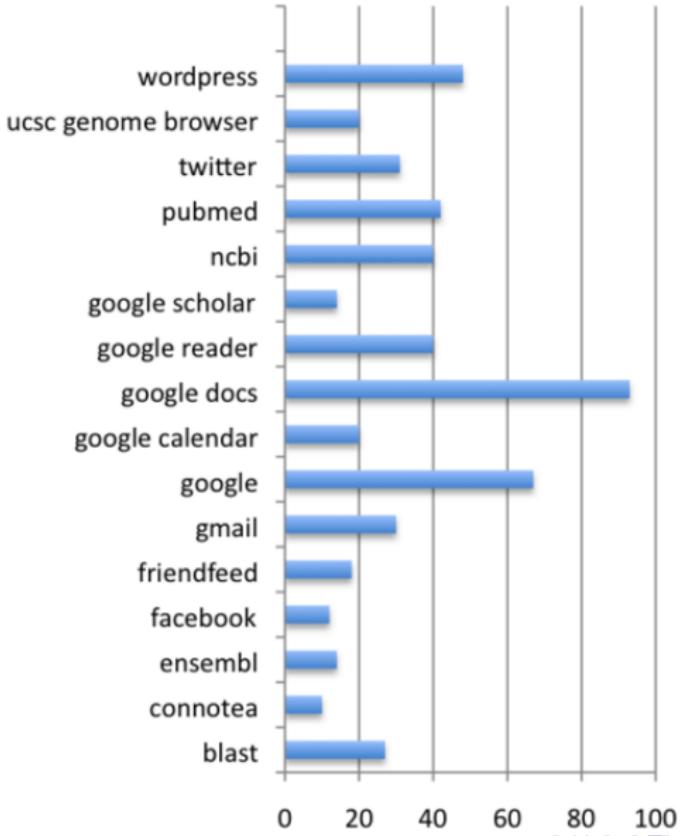


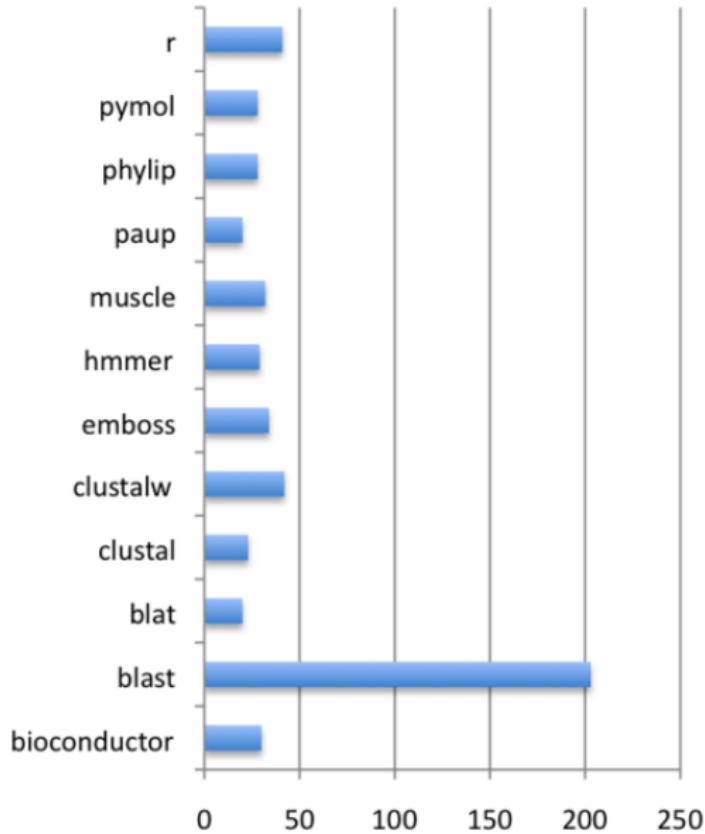
2012 | language





2008 | applications





industry vs. academic

- Academic Salary Mean/Median: \$36,520 / \$33,712
- Industry Salary Mean/Median: \$66,239 / \$64,235

Salary/Years Experience

- Academic Mean/Median: \$10,970 / \$8,333
- Industry Mean/Median: \$17,410 / \$12,000



industry vs. academic

- Academic Salary Mean/Median: \$36,520 / \$33,712
- Industry Salary Mean/Median: \$66,239 / \$64,235

Salary/Years Experience

- Academic Mean/Median: \$10,970 / \$8,333
- Industry Mean/Median: \$17,410 / \$12,000



- Bioinformatics Career Survey 2008
- Bioinformatics Career Survey 2008 Results
- bioinformatics-career-survey (GitHub)
- A comparison of common programming languages used in bioinformatics
- A comparison of bioinformatics programming languages
- Programming Languages of Bioinformatics
- A global perspective on evolving bioinformatics and data science training needs



Data and Bioinformatics Scientists (2018-26)

- The Future of Jobs
 - The top ten job winners
 - The top ten job losers
- What is the career outlook for Bioinformatics (2018-2026)?
 - What skills a successful bioinformatician need to have?
 - Specialization within a branch of bioinformatics is essential
 - Not all bioinformatics jobs and bioinformatics scientists are the same
 - Job growth and median wages
 - Keep an eye on rapid technology development
 - What technology knowledge employers demand today?
- Formal bioinformatics education
- Pros and cons of bioinformatics as an interdisciplinary field

教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



R

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

R 语言，一种自由软体程式语言与操作环境，主要用于**统计分析、绘图、数据挖掘**。R 本来是由来自新西兰奥克兰大学的罗斯·伊哈卡和罗伯特·杰特曼开发（也因此称为 R），现在由“R 开发核心团队”负责开发。

- R 内建多种统计学及数字分析功能。R 的功能也可以透过安装套件 (Packages, 用户撰写的功能) 增强。
- R 的另一强项是绘图功能，制图具有印刷的素质，也可加入数学符号。
- 虽然 R 主要用于统计分析或者开发统计相关的软体，但也有人用作矩阵计算。其分析速度可媲美专用于矩阵计算的自由软件 GNU Octave 和商业软件 MATLAB。

Packages

R 的功能能够透过由用户撰写的套件增强。增加的功能有特殊的统计技术、绘图功能，以及编程介面和数据输出/输入功能。这些软件包是由 R 语言、LaTeX、Java 及最常用 C 语言和 Fortran 撰写。

CRAN

CRAN 为 Comprehensive R Archive Network (R 综合典藏网) 的简称。它除了收藏了 R 的执行档下载版、原始码和说明文件，也收录了各种用户撰写的软件包。现时，全球有超过一百个 CRAN 镜像站。

Bioconductor

生物信息学社群时常使用 R 进行分子生物学数据分析。Bioconductor 计划就是让 R 作为基因图谱分析工具。



Packages

R 的功能能够透过由用户撰写的套件增强。增加的功能有特殊的统计技术、绘图功能，以及编程介面和数据输出/输入功能。这些软件包是由 R 语言、LaTeX、Java 及最常用 C 语言和 Fortran 撰写。

CRAN

CRAN 为 Comprehensive R Archive Network (R 综合典藏网) 的简称。它除了收藏了 R 的执行档下载版、原始码和说明文件，也收录了各种用户撰写的软件包。现时，全球有超过一百个 CRAN 镜像站。

Bioconductor

生物信息学社群时常使用 R 进行分子生物学数据分析。Bioconductor 计划就是让 R 作为基因图谱分析工具。

Packages

R 的功能能够透过由用户撰写的套件增强。增加的功能有特殊的统计技术、绘图功能，以及编程介面和数据输出/输入功能。这些软件包是由 R 语言、LaTeX、Java 及最常用 C 语言和 Fortran 撰写。

CRAN

CRAN 为 Comprehensive R Archive Network (R 综合典藏网) 的简称。它除了收藏了 R 的执行档下载版、原始码和说明文件，也收录了各种用户撰写的软件包。现时，全球有超过一百个 CRAN 镜像站。

Bioconductor

生物信息学社群时常使用 R 进行分子生物学数据分析。Bioconductor 计划就是让 R 作为基因图谱分析工具。

RStudio

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

RCode

A modern environment for R.



RStudio

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

RCode

A modern environment for R.



Useful and popular packages: Load data

readr readr makes it easy to read many types of **tabular data** including: delimited files, fixed width files and web log files.

readxl Import **excel files** into R. readxl supports both the legacy .xls format and the modern xml-based .xlsx format.

XLConnect Help you read, write and format **Micorsoft Excel** files from R.

foreign Functions for reading and writing data stored by some versions of Epi Info, Minitab, S, SAS, SPSS, Stata, Systat and Weka and for reading and writing some dBase files.

httr A set of useful tools for working with **http** connections.

XML Read and create **XML** documents with R.

jsonlite Read and create **JSON** data tables with R.

Useful and popular packages: Manipulate data

tidyverse Tools for changing the layout of your data sets. Use the gather and spread functions to convert your data into the **tidy format**, the layout R likes best.

magrittr magrittr provides a mechanism for chaining commands with a new **forward-pipe operator**, `%>%`.

dplyr Essential shortcuts for subsetting, summarizing, rearranging, and joining together data sets for **fast data manipulation**.



Useful and popular packages: Manipulate data

- lubridate** Tools that make working with **dates and times** easier.
- stringr** Easy to learn tools for **regular expressions and character strings**.
- plyr** plyr is a set of tools for a common set of problems: you need to **split** up a big data structure into homogeneous pieces, **apply** a function to each piece and then **combine** all the results back together.
- DT** The R package DT provides an R interface to the JavaScript library DataTables. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and DataTables provides filtering, pagination, sorting, and many other features in the tables.

Useful and popular packages: Visualize data

ggplot2 R's famous package for making beautiful graphics.

ggplot2 lets you use **the grammar of graphics** to build layered, customizable plots.

ggvis **Interactive, web based graphics** built with the grammar of graphics.

DiagrammeR Create **graph diagrams and flowcharts** using R.

htmlwidgets The htmlwidgets package provides a framework for easily creating R bindings to JavaScript libraries.



Useful and popular packages: Report results

- rmarkdown** rmarkdown lets you insert R code into a **markdown document**. R then generates a final document, in a wide variety of formats, that replaces the R code with its results.
- knitr** knitr is an elegant, flexible and fast dynamic report generation that combines R with TeX, Markdown, or HTML. For open access publishing, and **reproducible research** in statistics.
- xtable** The xtable function takes an R object (like a data frame) and returns the latex or HTML code you need to paste a pretty version of the object into your documents. Copy and paste, or pair up with R Markdown.

Useful and popular packages: Report results

Shiny Easily make **interactive, web apps** with R. A perfect way to explore data and share findings with non-programmers.

shinydashboard shinydashboard makes it easy to use Shiny to create dashboards.



Useful and popular packages: High performance

- data.table** An alternative way to organize data sets for **very, very fast** operations. Useful for big data.
- parallel** Use **parallel processing** in R to speed up your code or to crunch large data sets.
- foreach** Provides Foreach Looping Construct for R. Foreach is an idiom that allows for iterating over elements in a collection, without the use of an explicit loop counter. Using foreach without side effects also facilitates **executing the loop in parallel**.
- doParallel** Foreach Parallel Adaptor for the ‘parallel’ Package. Provides a parallel backend for the **%dopar%** function using the parallel package.

Useful and popular packages: Development

- devtools** An essential suite of tools for turning your code into an R package.
- roxygen2** A quick way to **document** your R packages. roxygen2 turns inline code comments into documentation pages and builds a package namespace.
- testthat** testthat provides an easy way to write **unit tests** for your code projects.



Easily install and load packages from the tidyverse

The **tidyverse** is a collection of R packages that share common philosophies and are designed to work together.

The **core tidyverse packages** that you are likely to use in almost every analysis:

- `readr`, for data import.
- `tidyr`, for data tidying.
- `dplyr`, for data manipulations.
- `ggplot2`, for data visualisation.
- `stringr`, for string manipulations.
- `purrr`, for functional programming.
- `tibble`, for tibbles, a modern re-imagining of data frames.
- `forcats`, a suite of useful tools that solve common problems with factors.

Easily install and load packages from the tidyverse

It also installs a selection of other tidyverse packages that you're likely to use frequently, but probably not in every analysis. This includes packages for:

- Working with specific types of vectors: hms (times), stringr (strings), lubridate (date/times),forcats (factors), blob (blob/binary data).
- Importing other types of data: DBI (databases), haven (SPSS, SAS and Stata files), httr (web apis), jsonlite (JSON), readxl (.xls and .xlsx files), rvest (web scraping), xml2 (XML).
- Programming and modelling: magrittr (%>%), glue (alternative to paste()), modelr (modelling within a pipeline), broom (turning models into tidy data).

Others

swirl Learn R, in R

learnr Interactive Tutorials for R

officer Manipulation of Microsoft Word and PowerPoint Documents

export Streamlined Export of Graphs and Data Tables

argparse A command line parser to be used with Rscript to write "#!" shebang scripts that gracefully accept positional and optional **arguments** and automatically generate usage.



Others

- viridis** Use the color scales in this package to make plots that are pretty, better represent your data, easier to read by those with colorblindness, and print well in grey scale.
- ggthemes** Extra Themes, Scales and Geoms for ‘ggplot2’
 - ggsci** Scientific Journal and Sci-Fi Themed Color Palettes for ‘ggplot2’
 - gpubr** ‘ggplot2’ Based Publication Ready Plots
- ggthemr** Themes for ggplot2
- ggtech** ggplot2 tech themes, scales, and geoms
- bbplot** R package that helps create and export ggplot2 charts in the style used by the BBC News data team



Others

ggtree Visualization and annotation of **phylogenetic trees**.

ComplexHeatmap ComplexHeatmap package provides a highly flexible way to arrange multiple **heatmaps** and supports self-defined annotation graphics.

esquisse Explore and Visualize Your Data Interactively

rattle Graphical User Interface for Data Science in R

visreg Visualization of Regression Models

ggThemeAssist Add-in to Customize ‘ggplot2’ Themes

ggedit Interactive ‘ggplot2’ Layer and Theme Aesthetic Editor

ggridge A Grammar of Animated Graphics



Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development.

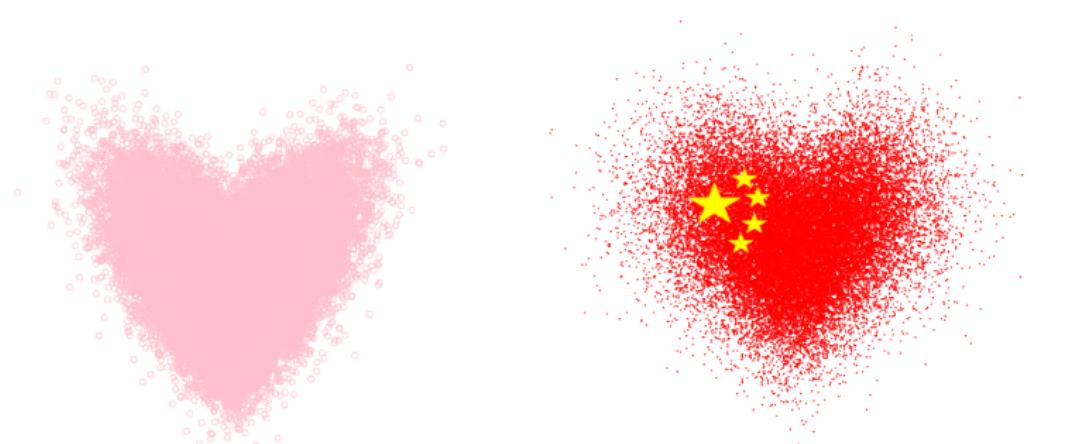


R | Example

```
1 n <- 50000
2 r <- 0.7
3 r_e <- (1 - r * r) ^ 0.5
4 X <- rnorm(n)
5 Y <- X * r + r_e * rnorm(n)
6 Y <- ifelse(X>0, Y, -Y)
7 plot(X, Y, col="pink")
```



R | Example



A dense, colorful cloud of text where the word "LOVE" is repeated in various colors (purple, blue, red, yellow, green) across the page.

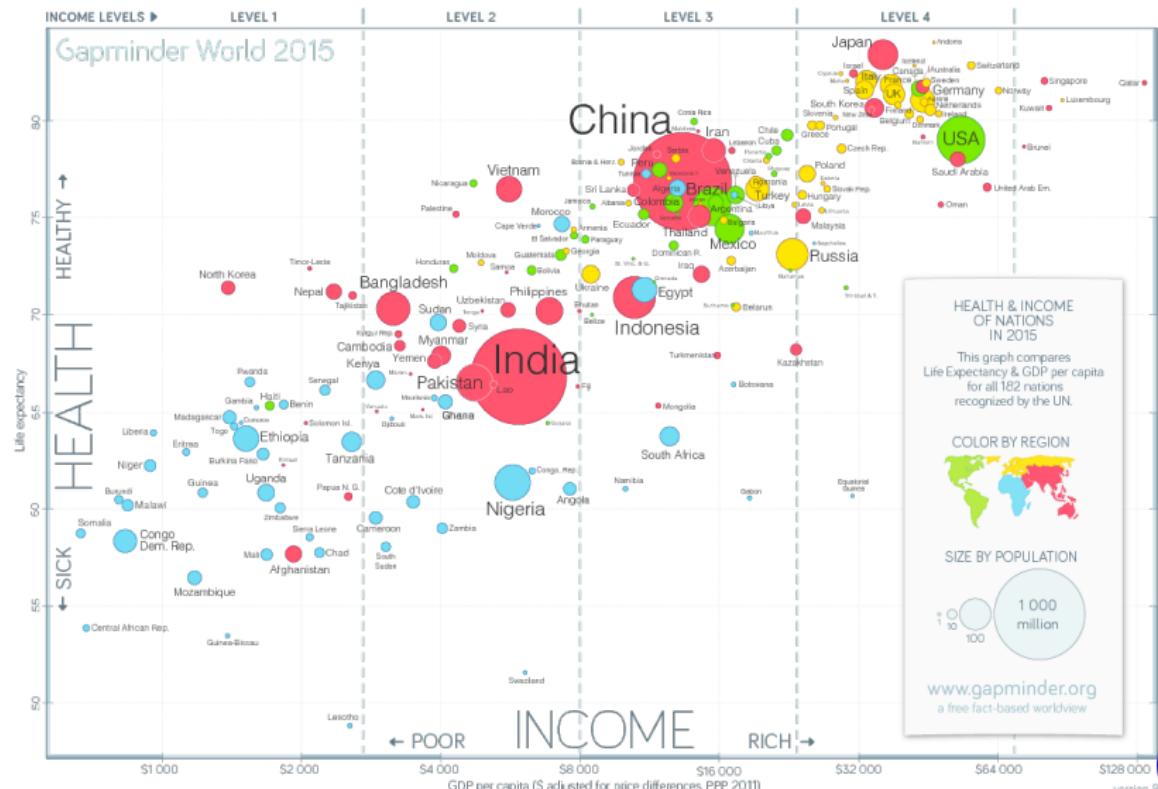


R | Example

```
1 xrange <- c(-15, 15); yrange <- c(0, 16)
2 plot(0, xlim=xrange, ylim=yrange, type="n")
3
4 yr <- seq(yrange[1], yrange[2], len=50)
5 offsetFn <- function(y) { 2 * sin(0 + y/3) }
6 offset <- offsetFn(yr)
7 leftE <- function(y) { -10 - offsetFn(y) }
8 rightE <- function(y) { 10 + offsetFn(y) }
9 xp <- c(leftE(yr), rev(rightE(yr)))
10 yp <- c(yr, rev(yr))
11 polygon(xp, yp, col="#ffeecc", border=NA)
12
13 h <- 9
14 xt <- seq(0, rightE(h), len=100)
15 yt <- log(1 + log(1 + log(xt + 1)))
16 yt <- yt - min(yt); yt <- h * yt/max(yt)
17 x <- c(leftE(h), rightE(h), rev(xt), -xt)
18 y <- c(h, h, rev(yt), yt)
19 polygon(x, y, col="red", border=NA)
```



R | Example



DATA SOURCES—INCOME: World Bank's GDP per capita, PPP 2011 (International \$) without additions by Gapminder. X-axis uses log scale to make a straight income line same distance on all levels. POPULATION: Numbers from UN Population Division. LIFE EXPECTANCY: <http://en.wikipedia.org/>—The interactive version of this chart is available at www.gapminder.org/tools, which lets you animate historic data for hundreds of indicators. LICENSE: Our charts are freely available under Creative Commons Attribution License. Please copy them, modify them and even sell them, as long as you mention "Based on a free chart from gapminder.org".



R | Example | parallel

```
1 #!/usr/bin/Rscript
2
3 # print 1 to 10: use 20 seconds
4 system.time(
5   for (i in 1:10) {
6     Sys.sleep(2)
7     print(i)
8   }
9 )
```



R | Example | parallel

```
1 #!/usr/bin/Rscript
2 library(foreach)
3 library(doParallel)
4
5 # no_cores <- detectCores() - 1
6 no_cores <- 10
7 cl <- makeCluster(no_cores)
8 registerDoParallel(cl)
9
10 system.time(
11   x <- foreach (i=1:10) %dopar% {
12     Sys.sleep(2); print(i)
13   }
14 )
15
16 stopCluster(cl)
```



- The R Project for Statistical Computing
- RStudio
- Learn R
- RStudio Cheat Sheets
- R packages inspired by R and its community
- Quick list of useful R packages
- Bioconductor

教学提纲

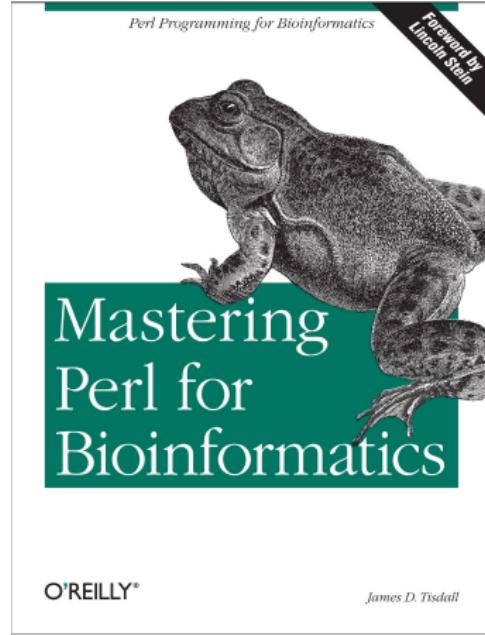
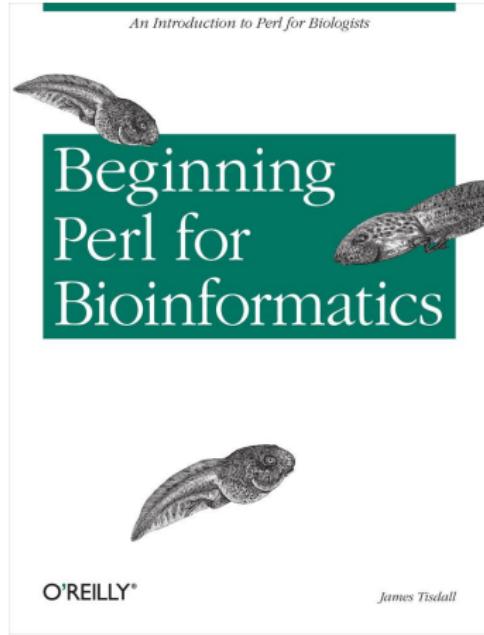
- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



生物信息学角度

Beginning \Longrightarrow *Mastering Perl for Bioinformatics*



编程语言角度：代号

小骆驼 ⇒ 羊驼书 ⇒ 小羊驼母亲和她的孩子 ⇒ 大骆驼 ⇒ 黑豹书

英文名

Learning Perl ⇒ *Intermediate Perl* ⇒ *Mastering Perl* ⇒ *Programming Perl*
⇒ *Advanced Perl Programming*

中文名

《Perl 语言入门》⇒ 《Perl 进阶》⇒ 《精通 Perl》⇒ 《Perl 语言编程》
⇒ 《高级 Perl 编程》



编程语言角度：代号

小骆驼 ⇒ 羊驼书 ⇒ 小羊驼母亲和她的孩子 ⇒ 大骆驼 ⇒ 黑豹书

英文名

Learning Perl ⇒ *Intermediate Perl* ⇒ *Mastering Perl* ⇒ *Programming Perl*
⇒ *Advanced Perl Programming*

中文名

《Perl 语言入门》⇒ 《Perl 进阶》⇒ 《精通 Perl》⇒ 《Perl 语言编程》
⇒ 《高级 Perl 编程》



编程语言角度：代号

小骆驼 ⇒ 羊驼书 ⇒ 小羊驼母亲和她的孩子 ⇒ 大骆驼 ⇒ 黑豹书

英文名

Learning Perl ⇒ *Intermediate Perl* ⇒ *Mastering Perl* ⇒ *Programming Perl*
⇒ *Advanced Perl Programming*

中文名

《Perl 语言入门》⇒ 《Perl 进阶》⇒ 《精通 Perl》⇒ 《Perl 语言编程》
⇒ 《高级 Perl 编程》



编程语言角度：代号

小骆驼 ⇒ 羊驼书 ⇒ 小羊驼母亲和她的孩子 ⇒ 大骆驼 ⇒ 黑豹书

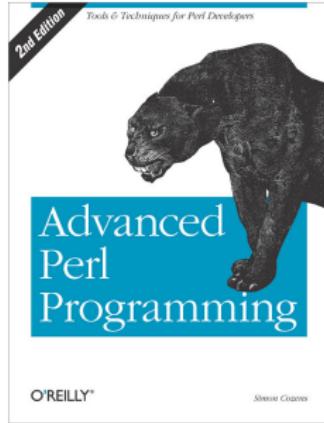
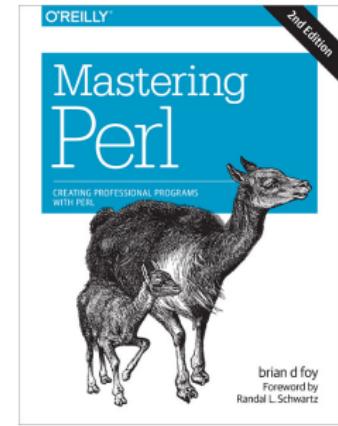
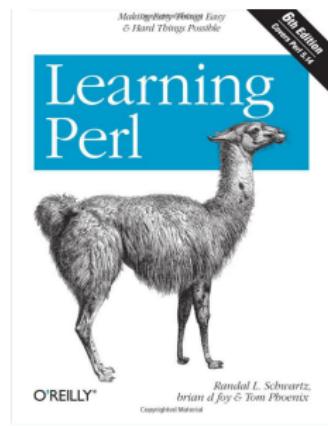
英文名

Learning Perl ⇒ *Intermediate Perl* ⇒ *Mastering Perl* ⇒ *Programming Perl*
⇒ *Advanced Perl Programming*

中文名

《Perl 语言入门》⇒ 《Perl 进阶》⇒ 《精通 Perl》⇒ 《Perl 语言编程》
⇒ 《高级 Perl 编程》





中文名

- 《Perl 入门经典》
- 《高阶 Perl》
- 《Perl 高效编程》
- 《Perl 最佳实践》
- *Perl Cookbook*

英文名

- *Beginning Perl*
- *Higher-Order Perl*
- *Effective Perl Programming*
- *Perl Best Practices*
- *Perl Cookbook*



书籍 | Perl



中文名

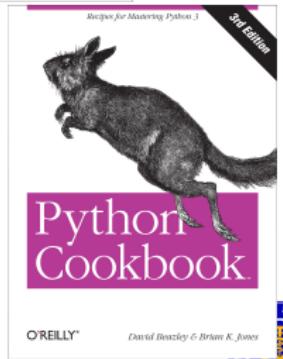
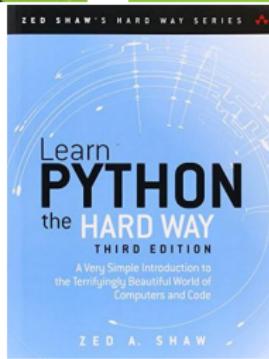
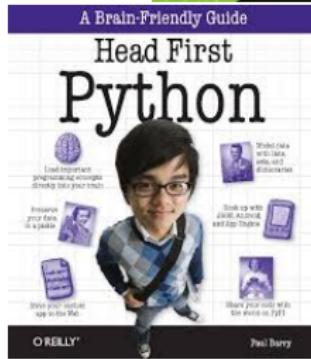
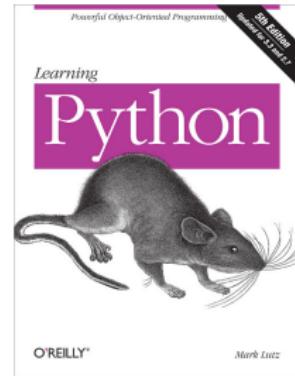
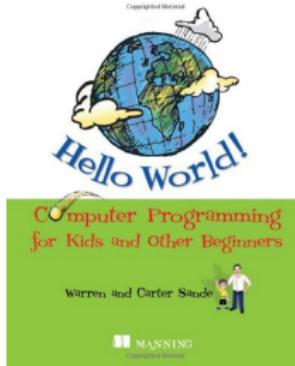
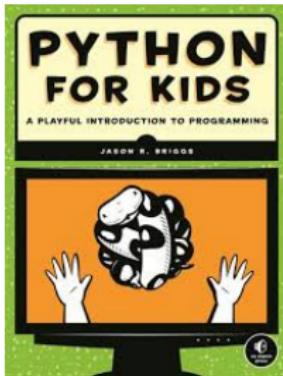
- 《趣学 Python 编程》
- 《父与子的编程之旅：与小卡特一起学 Python》
- 《Python 学习手册》
- 《深入浅出 Python》
- 《“笨办法”学 Python》
- 《像计算机科学家一样思考 Python》
- *Python Cookbook*

英文名

- *Python for Kids*
- *Computer Programming for Kids and Other Beginners*
- *Learning Python*
- *Head First Python*
- *Learn Python the Hard Way*
- *Think Python*
- *Python Cookbook*



书籍 | Python



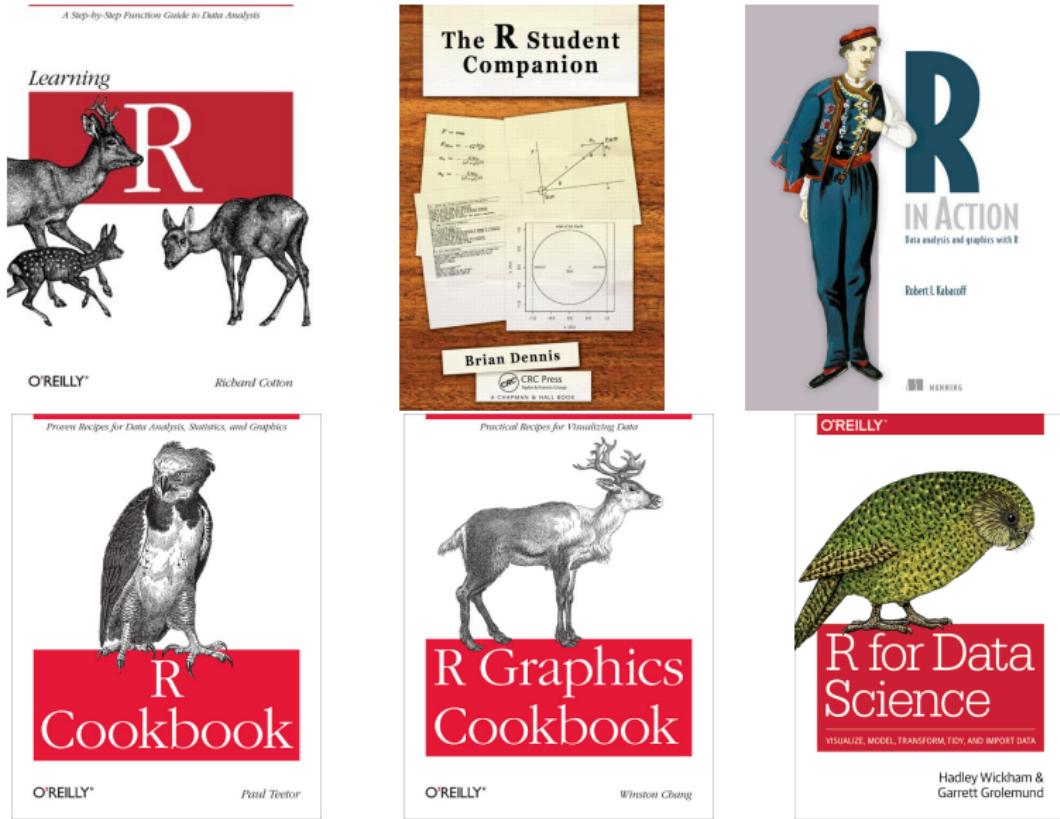
中文名

- 《学习 R》
- 《R 语言初学指南》
- 《R 语言实战》
- 《R 语言经典实例》
- 《R 数据可视化手册》
- 《R 数据科学》

英文名

- *Learning R*
- *The R Student Companion*
- *R in Action*
- *R Cookbook*
- *R Graphics Cookbook*
- *R for Data Science*





- 《Python 数据科学手册》
- 《Python 数据科学入门》
- 《Python 数据分析基础》
- 《Python 编程入门》
- 《Python 基础教程》
- 《Python 语言及其应用》
- 《Python 编程——从入门到实践》
- 《R 语言入门与实践》
- 《R 语言：从数据思维到数据实战》
- 《R 语言数据可视化之美：专业图表绘制指南》



计算机科学与程序设计

- 《Linux Shell 脚本攻略》
- 《我的第一本编程书》
- 《教孩子学编程》
- 《编程导论》
- 《深入浅出程序设计》
- 《程序是怎样跑起来的》
- 《写给大家看的算法书》
- 《啊哈！算法》
- 《算法的乐趣》
- 《算法帝国》
- 《算法笔记》
- 《轻松学算法》
- 《数据结构与算法图解》
- 《大话数据结构》
- 《程序员的数学》
- 《程序员的数学 2：概率统计》
- 《程序员的数学 3：线性代数》
- 《统计思维：程序员数学之概率统计》
- 《程序员的数学思维修炼》

数据科学

- 《数据科学入门》
- 《面向数据科学家的实用统计学》
- 《命令行中的数据科学》
- 《深入浅出数据分析》
- 《菜鸟侦探挑战数据分析》



统计学

- 《统计学》 (*Statistics for Engineers and the Sciences*)
- 《统计学及其应用》
- 《深入浅出统计学》
- 《从零开始学统计》
- 《从零开始读懂统计学》
- 《白话统计》
- 《白话统计学》
- 《爱上统计学》
- 《赤裸裸的统计学》
- 《你一定爱读的极简统计学》
- 《介绍丛书: 统计学》
- 《漫画玩转统计学》
- 《漫画统计学》



统计学科普

- 《统计学的世界》
- 《统计学漫话》
- 《统计学七支柱》
- 《数字唬人》
- 《统计数字会撒谎》
- 《统计数据的真相》
- 《生活中的概率趣事》
- 《改变世界的 134 个概率统计故事》
- 《统计会犯错——如何避免数据分析中的统计陷阱》
- 《妙趣横生的统计学——培养大数据时代的统计思维》
- 《别拿相关当因果——因果关系简易入门》

教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



知识点

- 生物信息学：交叉学科，多种技能，领域宽泛
- 生物学：DNA, RNA, 蛋白质
- 计算机科学：三个术语，编程语言
- R：readr、tidyr、dplyr、ggplot2 等常用包



教学提纲

- 1 课程安排
- 2 生物信息学
- 3 生物学
- 4 计算机科学
- 5 编程

- 6 Bioinformatics Career Survey
2008
- 7 R
- 8 书籍
- 9 回顾与总结
 - 总结
 - 思考题



- ① 生物信息学主要是由哪些学科交叉而来的？
- ② 生物信息学主要需要哪些方面的知识和技能？
- ③ DNA 是由哪四种碱基组成的？RNA 与之有何不同？
- ④ 列举常见的 20 种氨基酸，它们三字母和单字母的缩写分别是什么？
- ⑤ 列举常见的存储 DNA 序列和蛋白质结构的数据库。
- ⑥ *in vivo*、*in vitro* 和 *in silico* 分别代表什么含义？
- ⑦ 列举常见的编程语言，在生物信息学中常用的编程语言，专用于生物信息学的工具集。
- ⑧ 列举 R 的常用包并简介其主要用途。



Markdown

回顾、总结 Markdown 标记语言的基本语法：

- 标题
- 强调
- 列表
- 代码
- 引用
- 链接
-



Powered by



T_EX L^AT_EX X_ET_EX Beamer