

# 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

2016-2017 学年上学期 (秋)  
2014 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

2016 年 11 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 回顾与总结
  - 总结
  - 思考题

- 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 回顾与总结
  - 总结
  - 思考题





后 9 周，每周三，下午四节（13:30-17:30），西楼 610

## 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编写程序
- 任务主题不限（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：无需排除的元素）
- 挖掘更多人性化需求……



## 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：无需排除的元素）
- 挖掘更多人性化需求……



后 9 周，每周四，上午后两节（10:10-12:00），教一楼 304

## 实验内容

- 紧跟理论课进度
- 教材上的程序/实例



① 理论课：60%

- ① 平时表现：10%
- ② 闭卷考试：50%

② 实验课：40%

- ① 平时表现：20%
- ② 实验报告：20%



# 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 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);GTATTTCGTACATTACTGCCAGCCACCAGAATATTGTACGGTAC
A#print STDERR 'blast args: ', Dumper( \%args ), $/;A
CACCCACTAGGATACCAACAAACCTACCCACCCCTAACAGTACATAGTACATC
C#y $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
CTGTTCTTCATGGGGAAAGCAGATTTGGGTACCCACCAAGTATTGACCAACCA
C# 0 == successTACATTACTGCCAGCCACCAGAATATTGTACGGTAC
A# 2 == success, no resultsAAGCAAGTACAGCAATCAACCCCTATA
Cif ((0 == $rc || 2 == $rc) && -f $out_file) {GTACATC
CTGTTmy $alignment = '';TTGGGTACCCACCAAGTATTGACCAACCA
CCGCTif ($fh->open($out_file)) {CCATGAATATTGTACGGTAC
ATATCAAAwhile (<$fh>) {TACAAGCAAGTACAGCAATCAACCCCTATA
CACCCACTAGGAT$alignment .= $_;CCCTTAACAGTACATAGTACATC
CTGTTCTT}ATGGGGAAAGCAGATTTGGGTACCCACCAAGTATTGACCAACCA
CCGCTATGT$fh->close;TACTGCCAGCCACCAGAATATTGTACGGTAC
ATATC)AAACCCCCCTCCCCATGCTTACAAGCAAGTACAGCAATCAACCCCTATA
CACCC$blast = DNALC::Pipeline::Phylogenetics::Blast->
CACCCACTAGGATproject_id => $self->project->id,GTACATC
```



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。目前的生物信息学基本上只是分子生物学与信息技术（尤其是互联网技术）的结合体。

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、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。



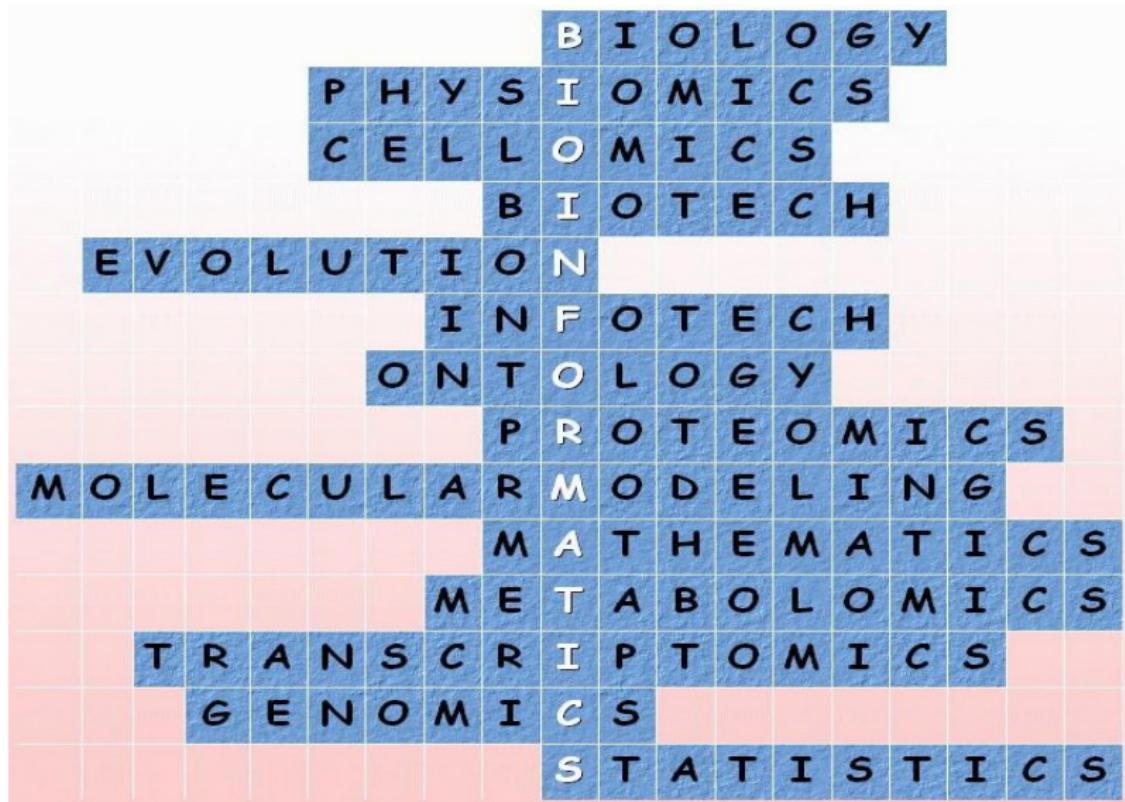
生物信息学可以定义为对分子生物学中两类信息流的研究：

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。

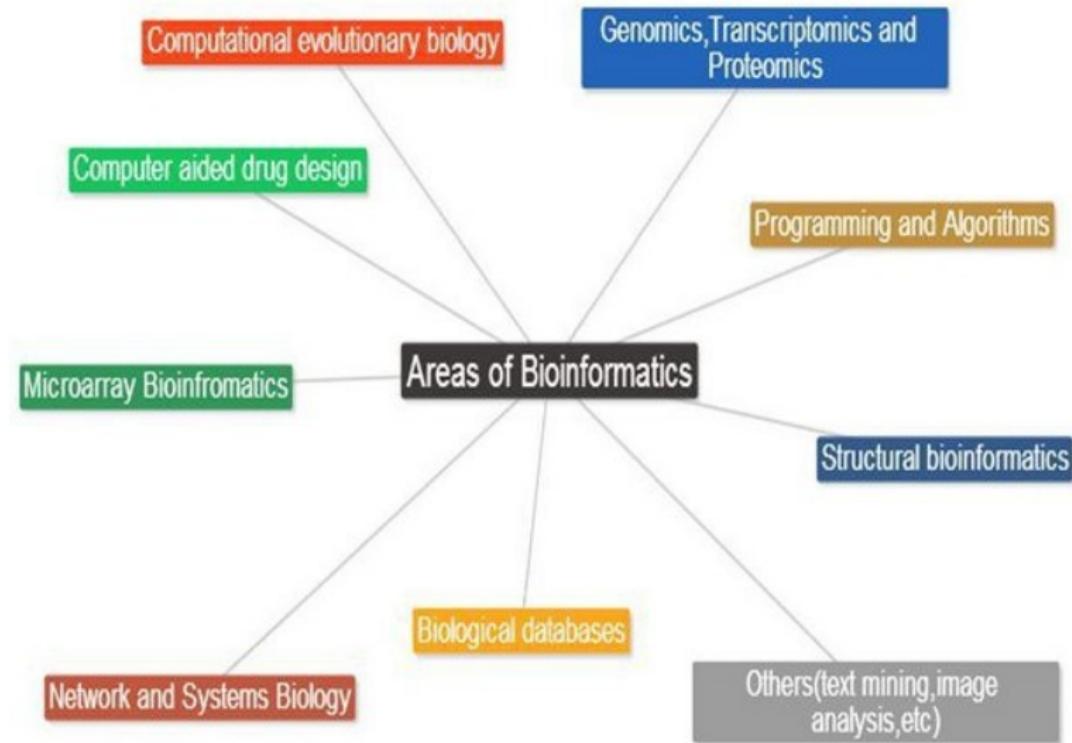
第二类信息流是基于科学方法：提出关于生物学活动的假设，设计实验以验证这些假设，评估结果与假设的相容性，然后根据实验数据对原假设作扩展或修正。第二类的生物信息学应用关注于这一流程中的信息传递，包括产生假设、设计实验、通过数据库将实验结果组织起来、检验数据与模型的相容性以及修正假设的各个系统。

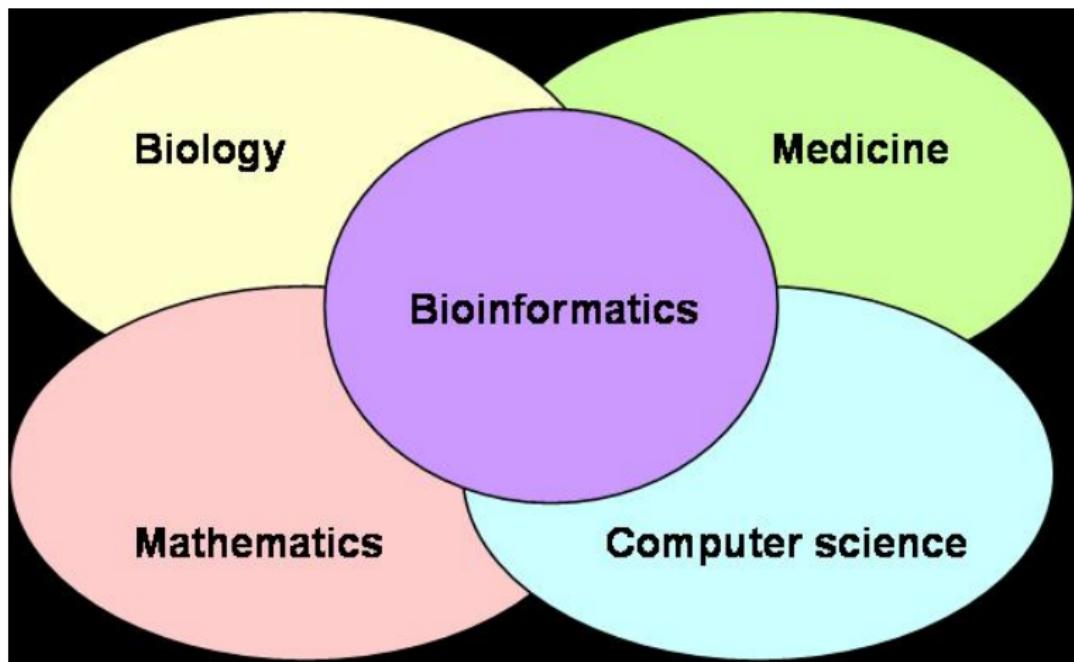


# 生物信息学 | 是什么? | 学科角度

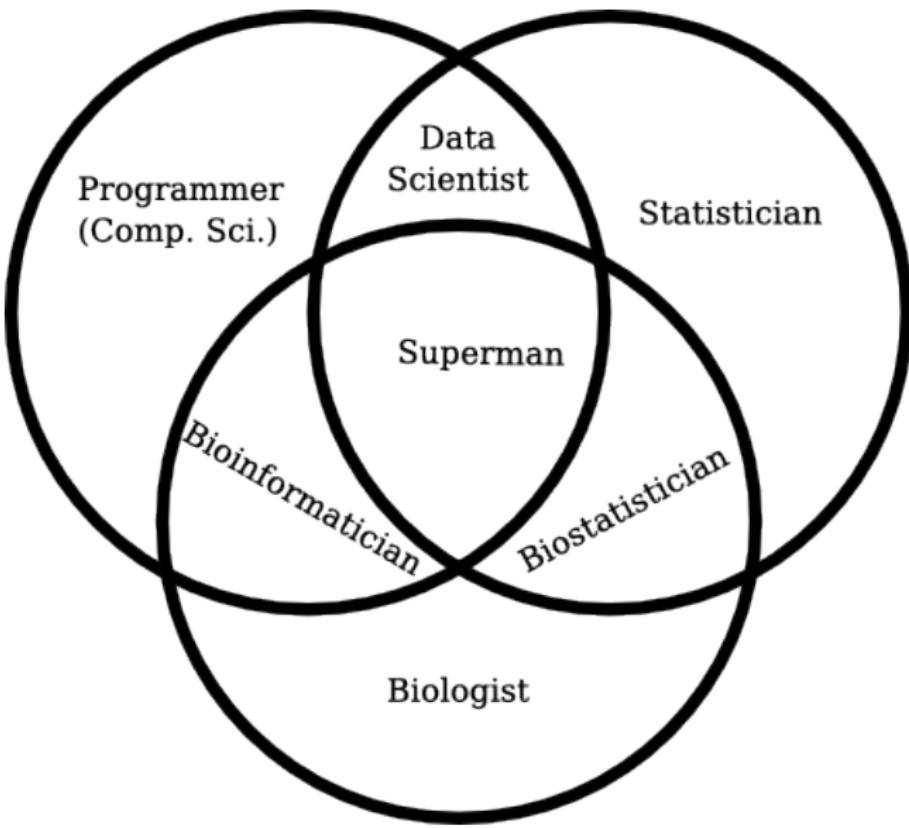


# 生物信息学 | 是什么? | 学科角度

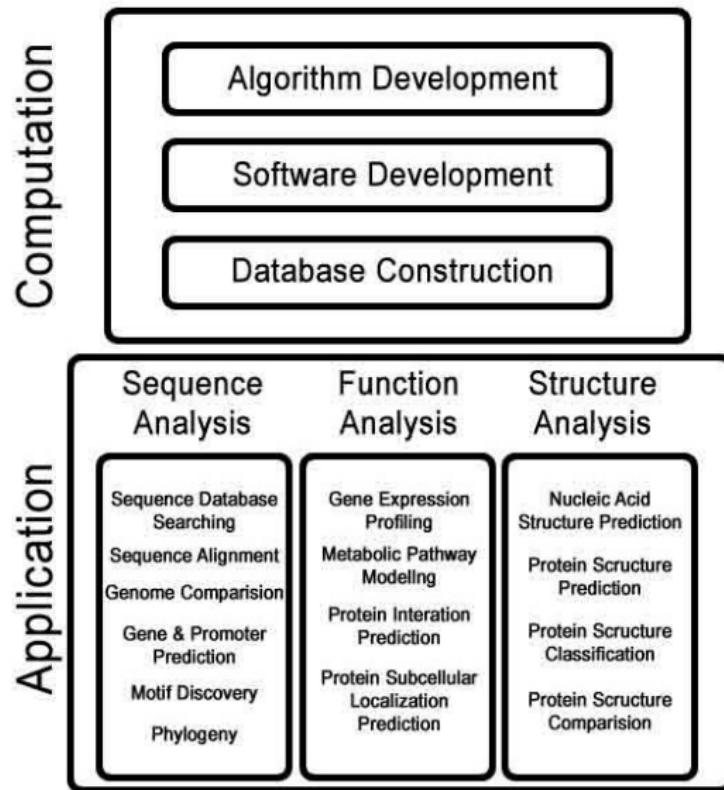




# 生物信息学 | 是什么? | 技术角度



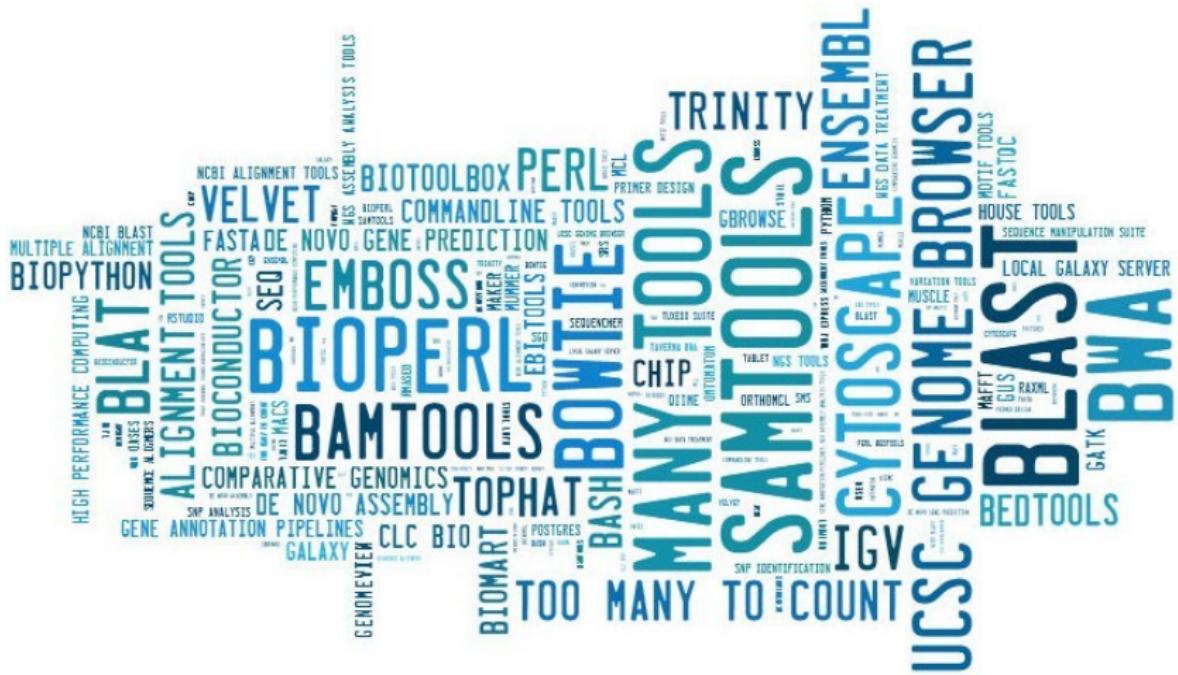
# 生物信息学 | 做什么?



# 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 | 主题词



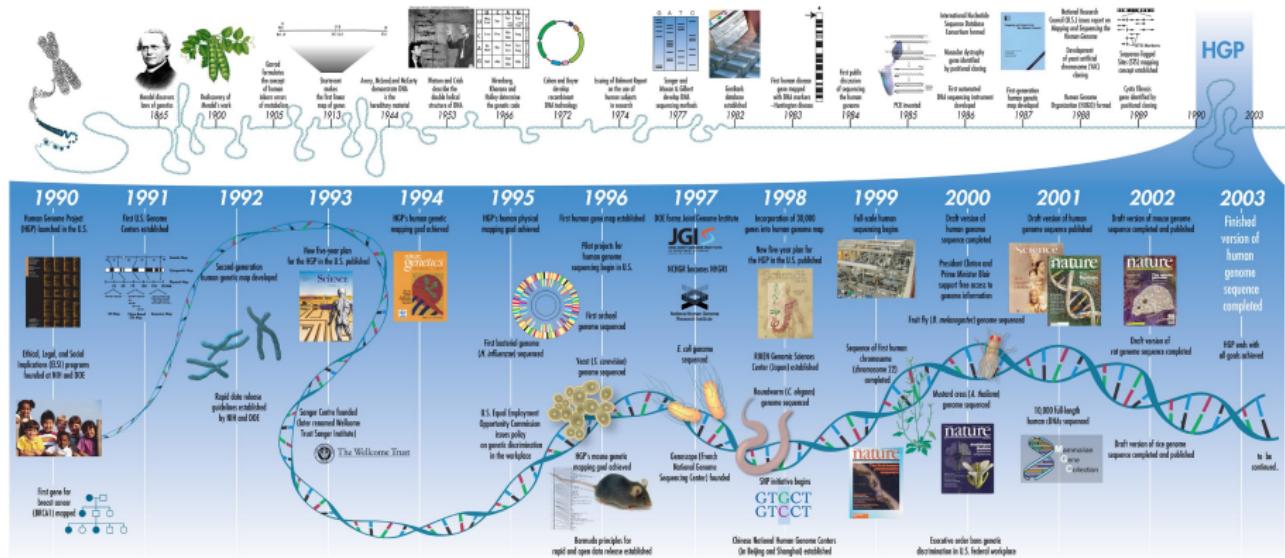
# 教学提纲

- 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 回顾与总结
  - 总结
  - 思考题



# 生物学 | 历史



## Central Dogma: DNA → RNA → Protein



DNA

transcription

CCTGAGCCAACCTATTGATGAA

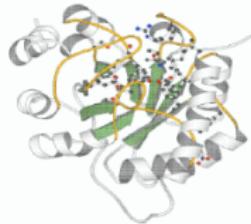
RNA

translation

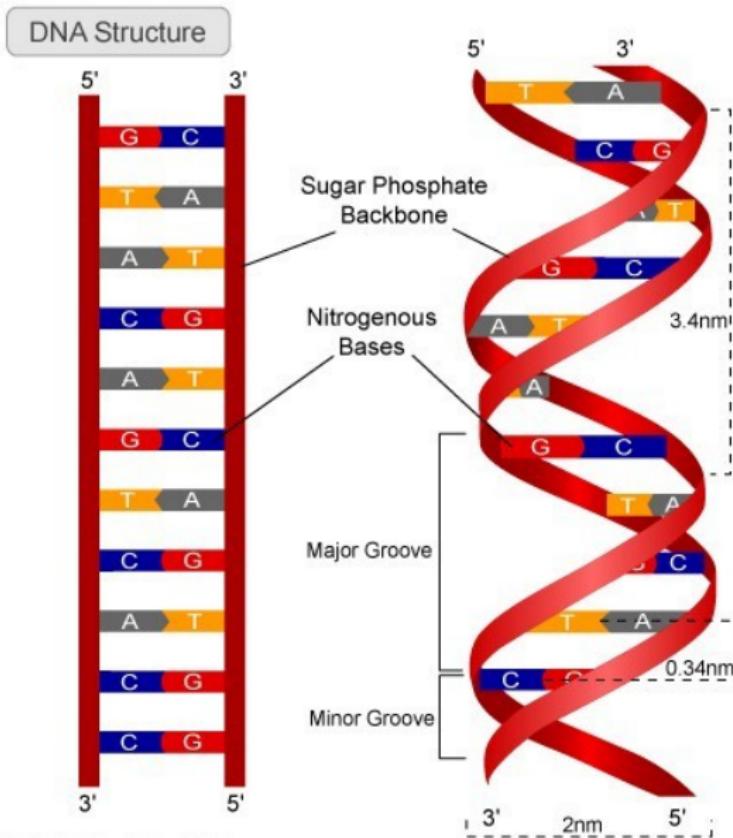
CCUGAGCCAACUAUUGAUGAA

Protein

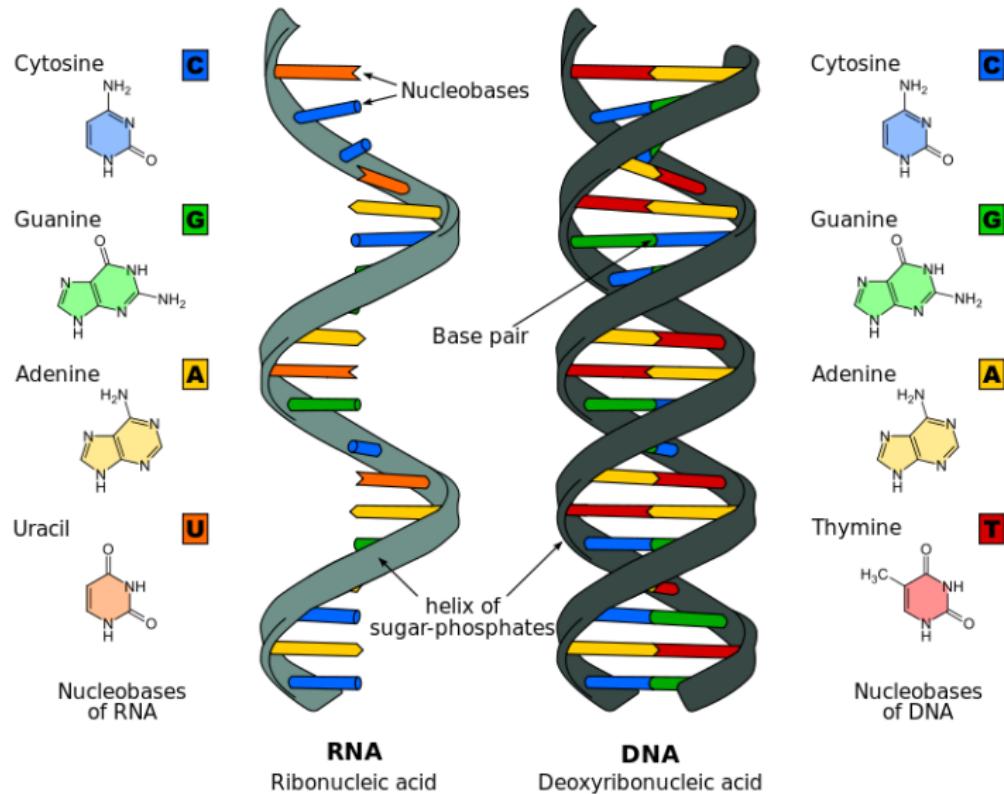
PEPTIDE



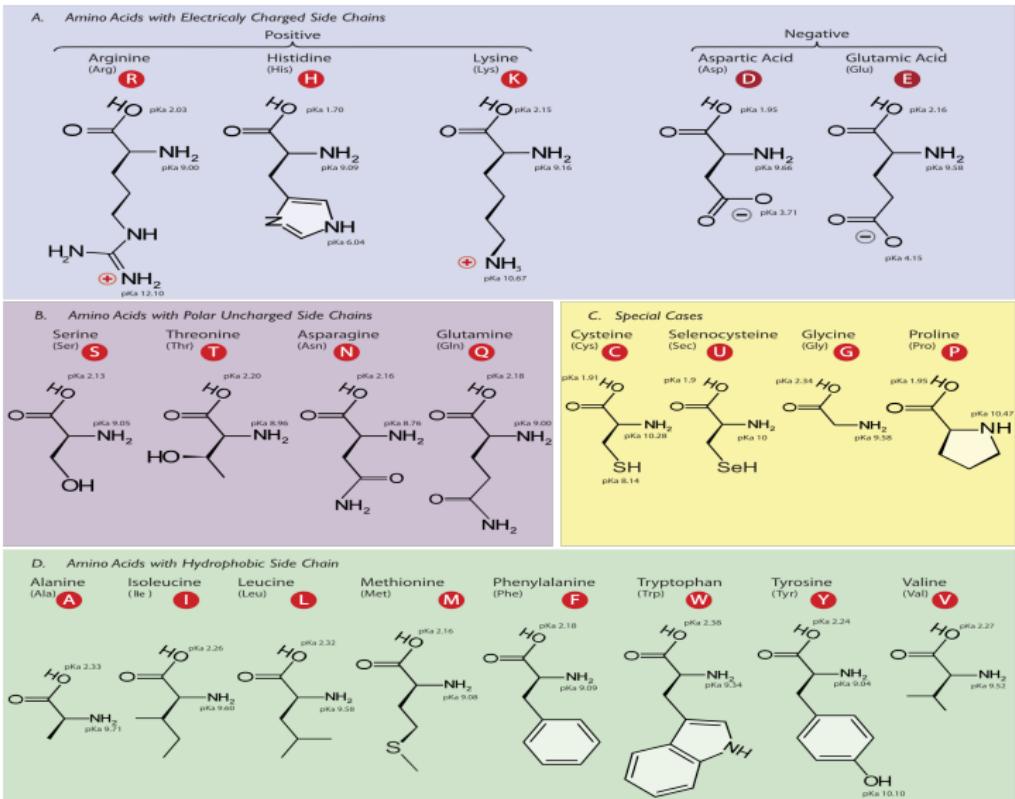
# 生物学 | DNA



# 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，20 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)

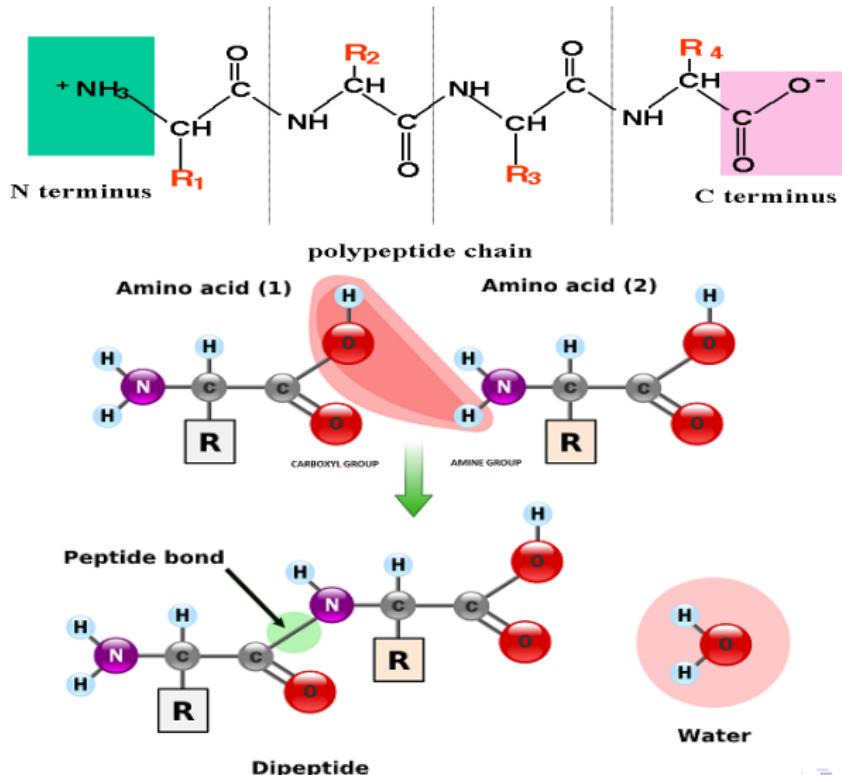
丙组丝甘半，天谷建酸胺。

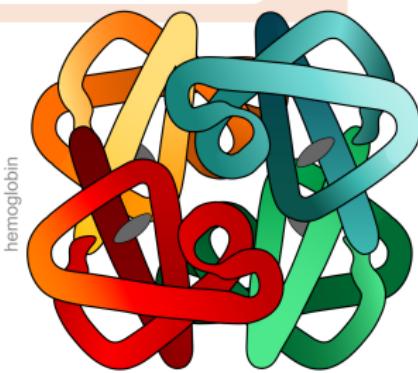
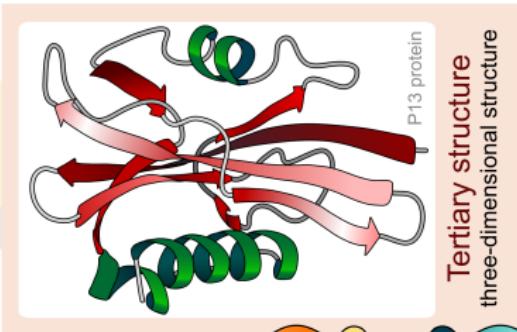
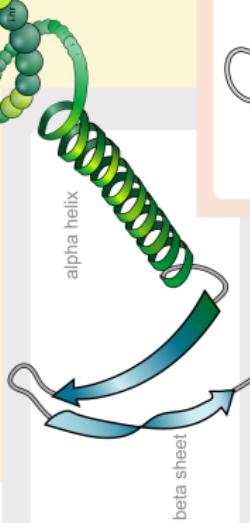
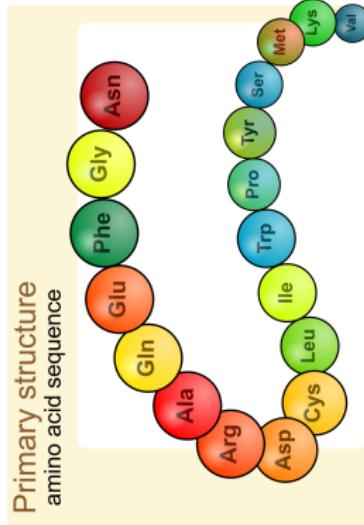
精酪加一脯，20 氨基酸。

(以上是非必需氨基酸)



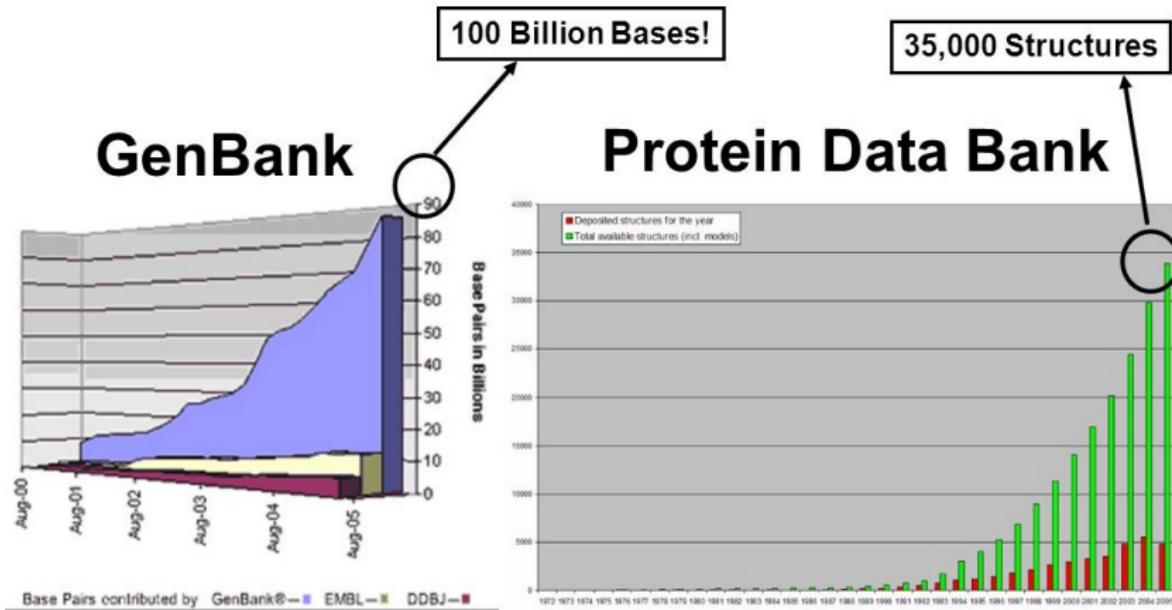
## Peptide = chain of amino acids





Quaternary structure  
complex of protein molecules



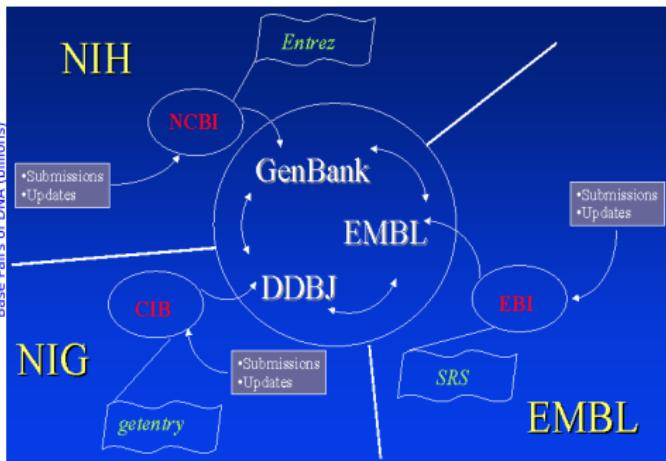
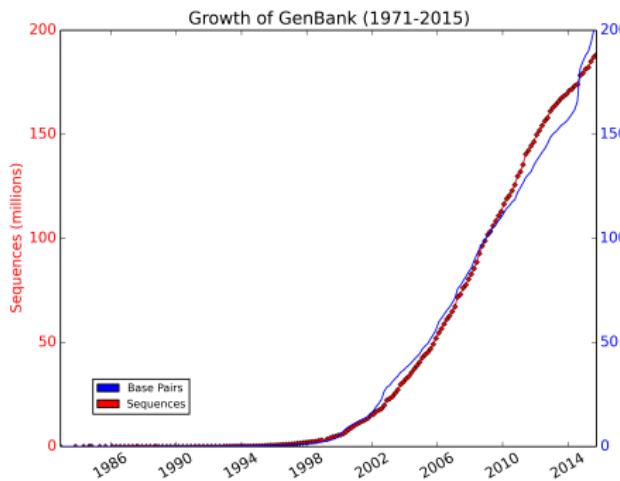


Base Pairs contributed by GenBank® — ■ EMBL — ■ DDBJ — ■

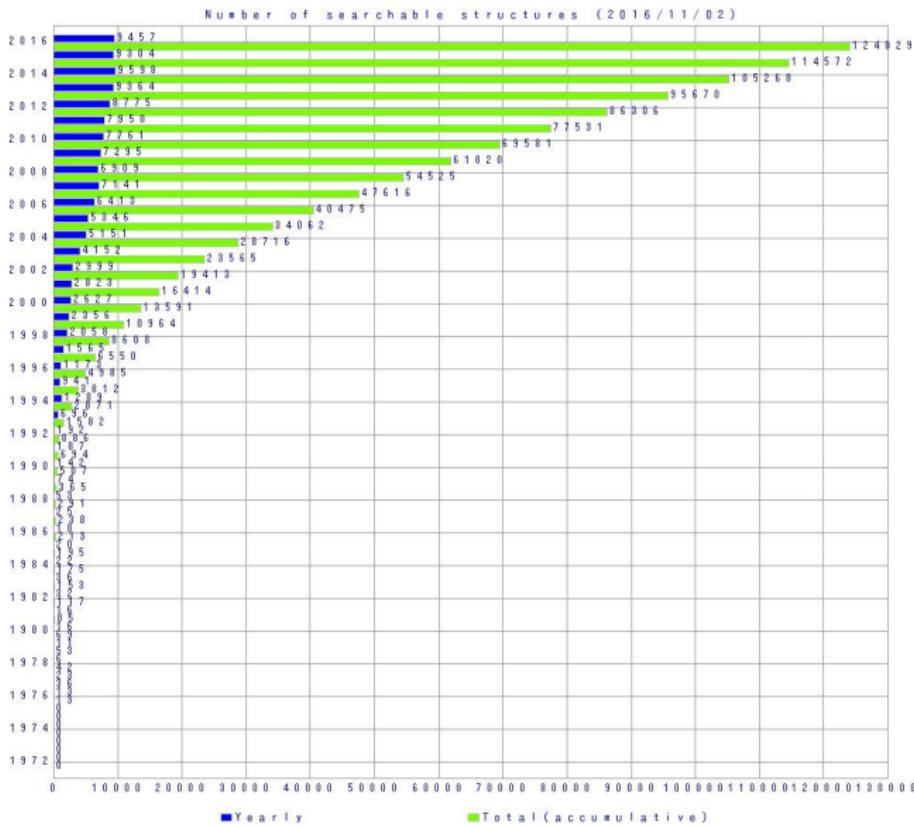
[www.ncbi.nlm.nih.gov/Genbank](http://www.ncbi.nlm.nih.gov/Genbank)

[www.rcsb.org/pdb/holdings.html](http://www.rcsb.org/pdb/holdings.html)





# 生物学 | 数据库 | PDB



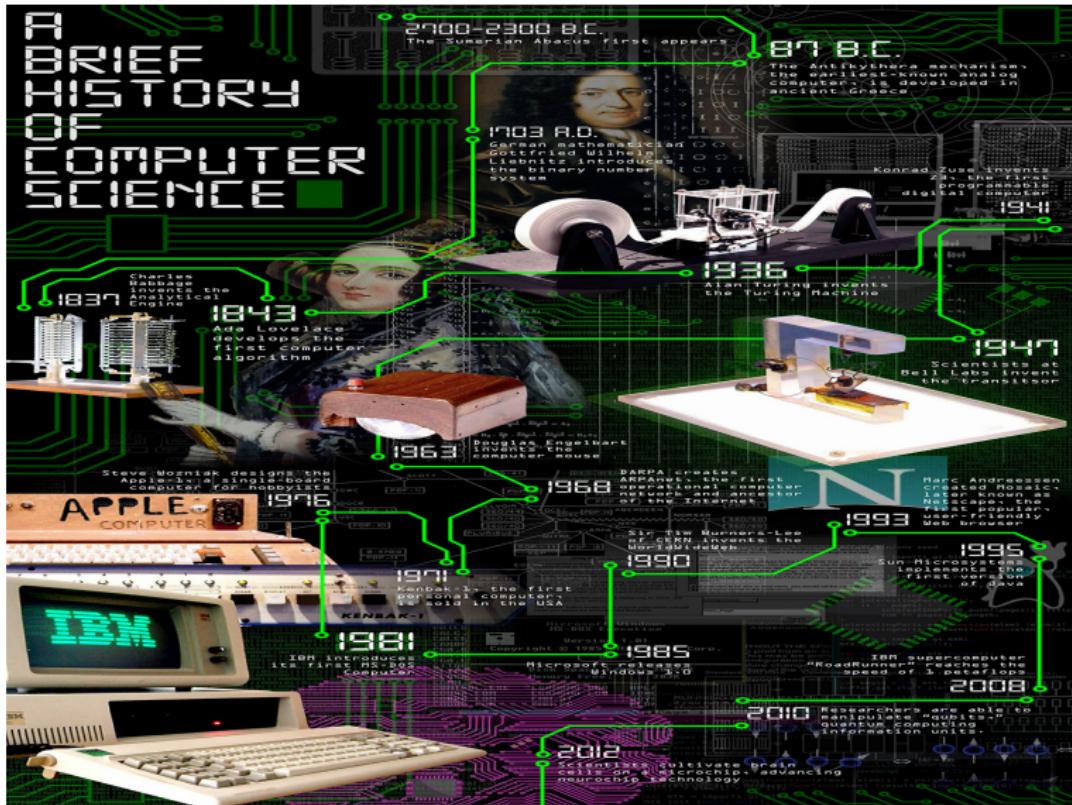


# 教学提纲

- 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 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。

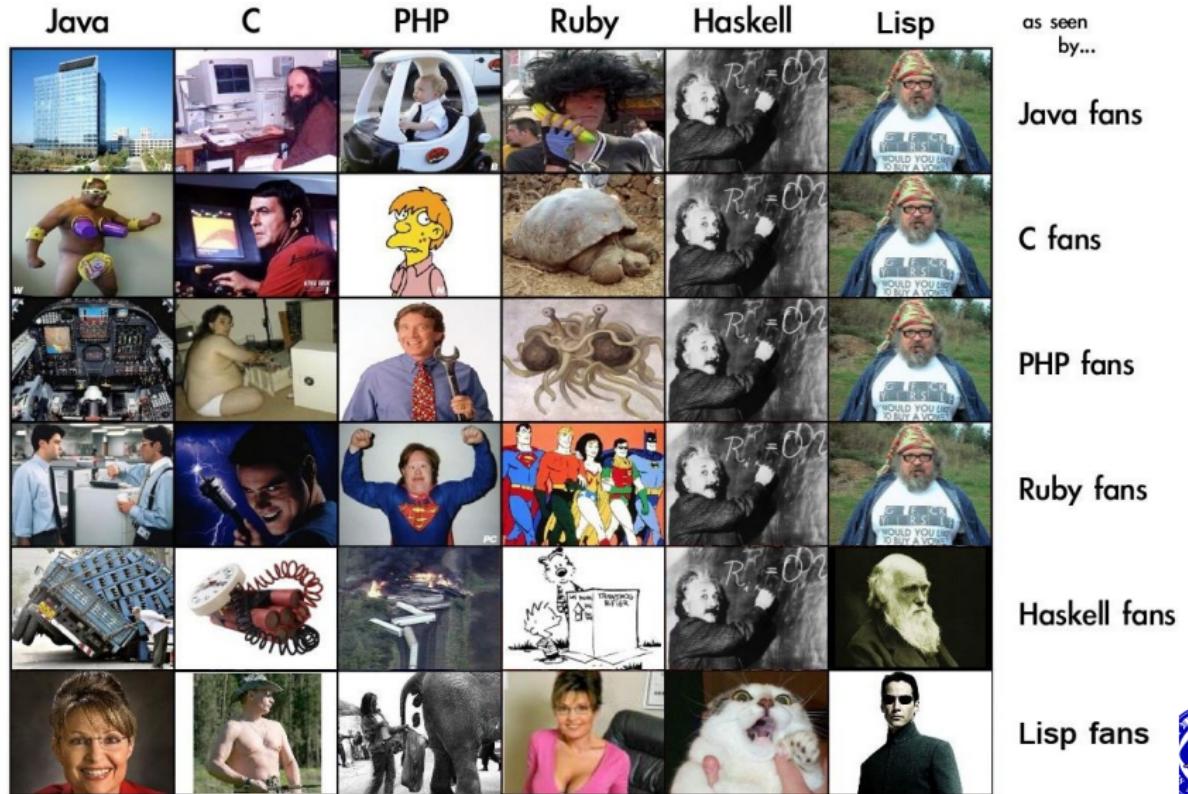
## *in silico*

*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

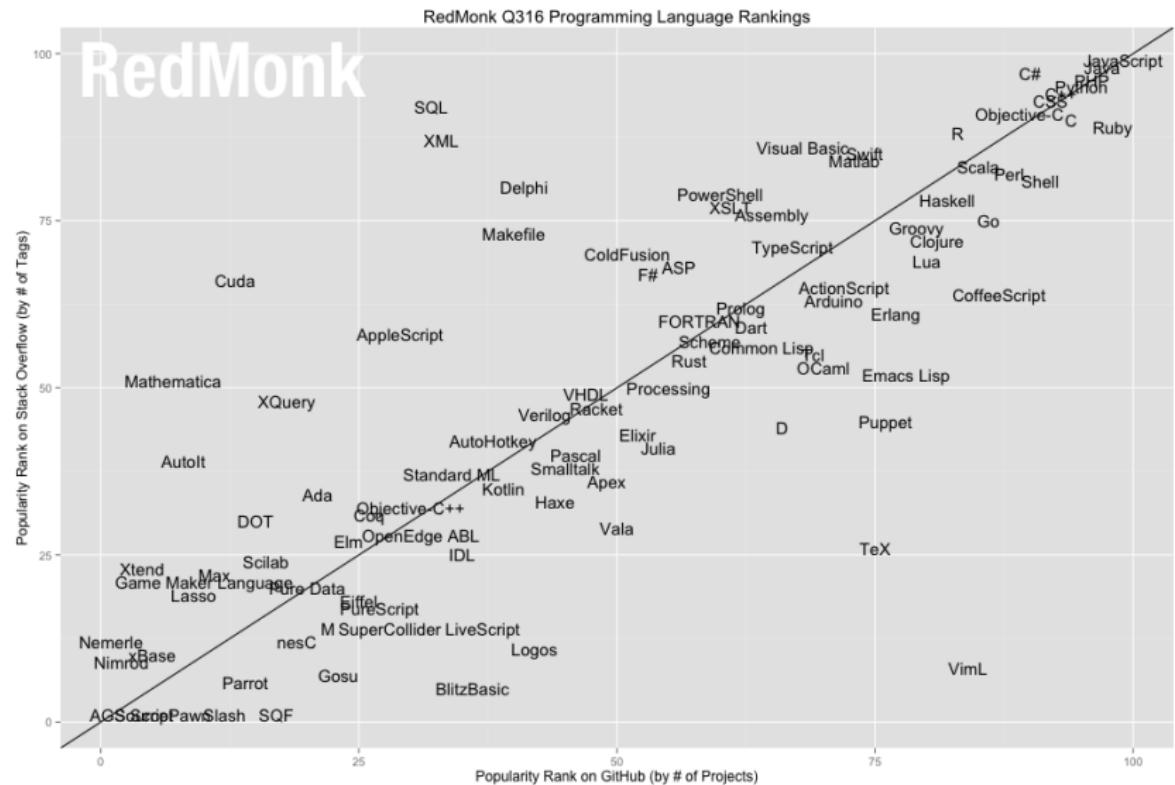
	C++		JavaScript
	Java/C#		PHP(Without MySQL)
	Ruby		Pascal
	Perl		Lisp
	Visual Basic		Haskell
	Python		C

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



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	—





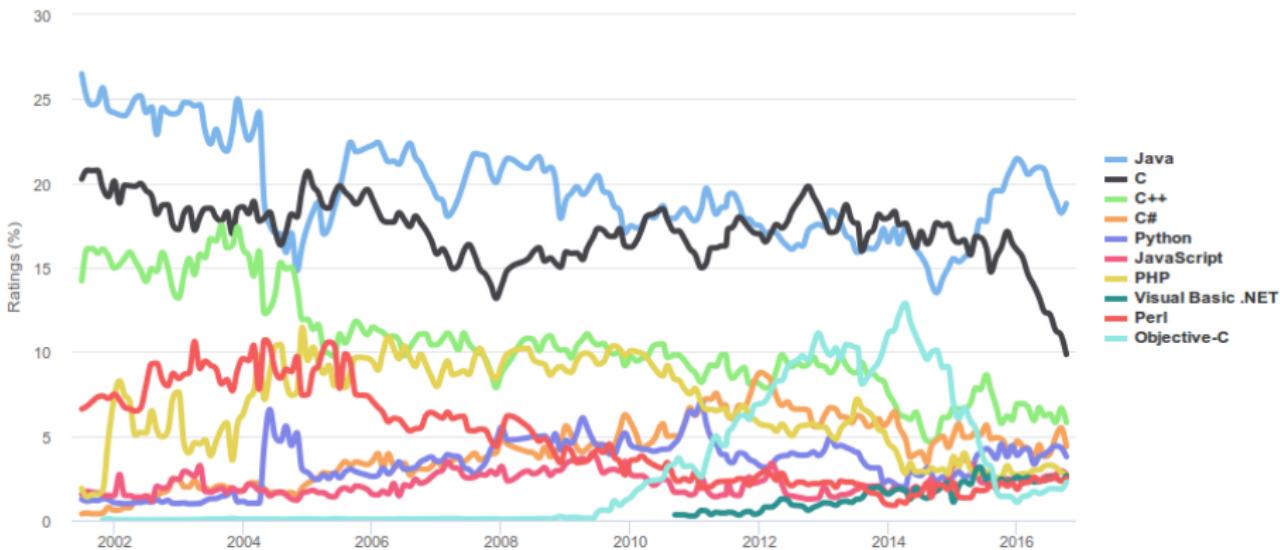
# 计算机科学 | 编程语言 | 排名 | 2016

Oct 2016	Oct 2015	Change	Programming Language	Ratings	Change
1	1		Java	18.799%	-0.74%
2	2		C	9.835%	-6.35%
3	3		C++	5.797%	+0.05%
4	4		C#	4.367%	-0.46%
5	5		Python	3.775%	-0.74%
6	8	▲	JavaScript	2.751%	+0.46%
7	6	▼	PHP	2.741%	+0.18%
8	7	▼	Visual Basic .NET	2.660%	+0.20%
9	9		Perl	2.495%	+0.25%
10	14	▲	Objective-C	2.263%	+0.84%
11	12	▲	Assembly language	2.232%	+0.66%
12	15	▲	Swift	2.004%	+0.73%
13	10	▼	Ruby	2.001%	+0.18%
14	13	▼	Visual Basic	1.987%	+0.47%
15	11	▼	Delphi/Object Pascal	1.875%	+0.24%
16	65	▲	Go	1.809%	+1.67%
17	32	▲	Groovy	1.769%	+1.19%
18	20	▲	R	1.741%	+0.75%
19	17	▼	MATLAB	1.619%	+0.46%
20	18	▼	PL/SQL	1.531%	+0.46%



## TIOBE Programming Community Index

Source: www.tiobe.com



# 计算机科学 | 编程语言 | 排名 | 历史

Programming Language	2016	2011	2006	2001	1996	1991	1986
Java	1	1	1	2	15	-	-
C	2	2	2	1	1	1	1
C++	3	3	3	3	2	2	5
C#	4	5	6	10	-	-	-
Python	5	6	7	23	24	-	-
PHP	6	4	4	8	-	-	-
JavaScript	7	9	8	7	19	-	-
Visual Basic .NET	8	28	-	-	-	-	-
Perl	9	8	5	4	3	-	-
Ruby	10	10	17	31	-	-	-
Lisp	27	12	12	15	7	4	3
Ada	28	16	15	16	6	5	2
Pascal	74	14	16	13	4	3	6



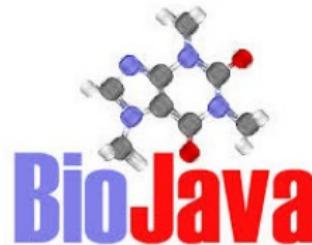
## 生物信息学常用编程语言

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



## 生物信息学专用

BioPerl、Biopython、BioRuby、BioJava



- 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 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，物价涨如赤兔
- .....



## 不需要！

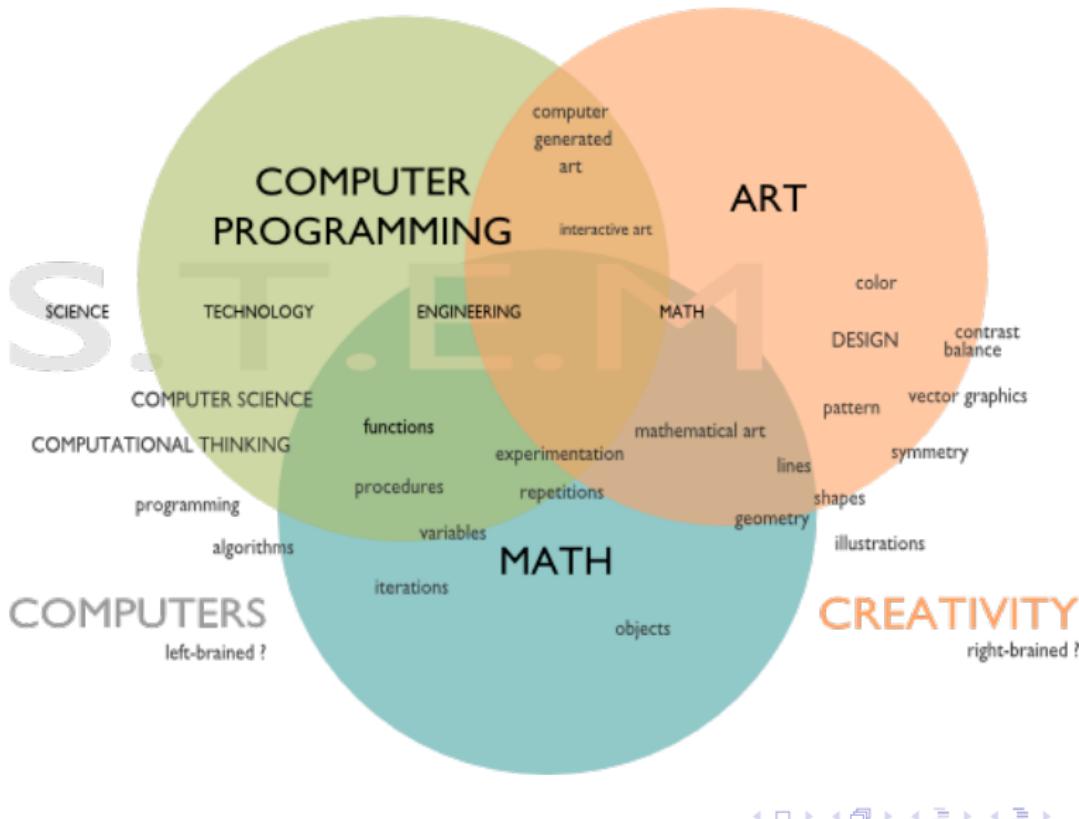
- 有现成的工具可以使用
- 五湖四海皆兄弟
- 有钱能使磨推鬼
- .....

## 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，物价涨如赤兔
- .....



- 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
- 人生苦短，学习编程
- .....

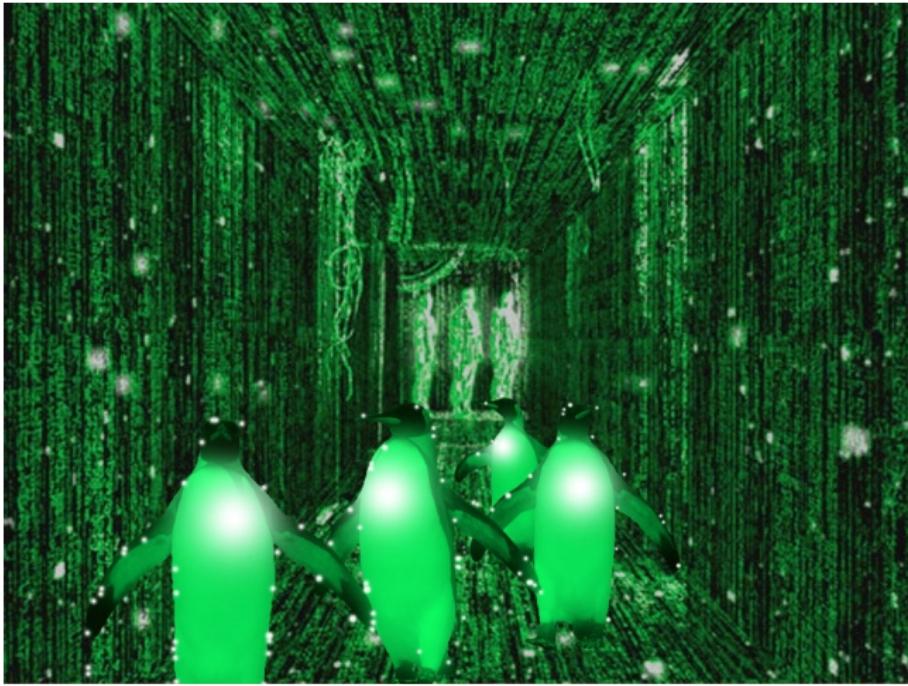


编程 | 与艺术

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】
- 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

## 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】
- 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

## 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

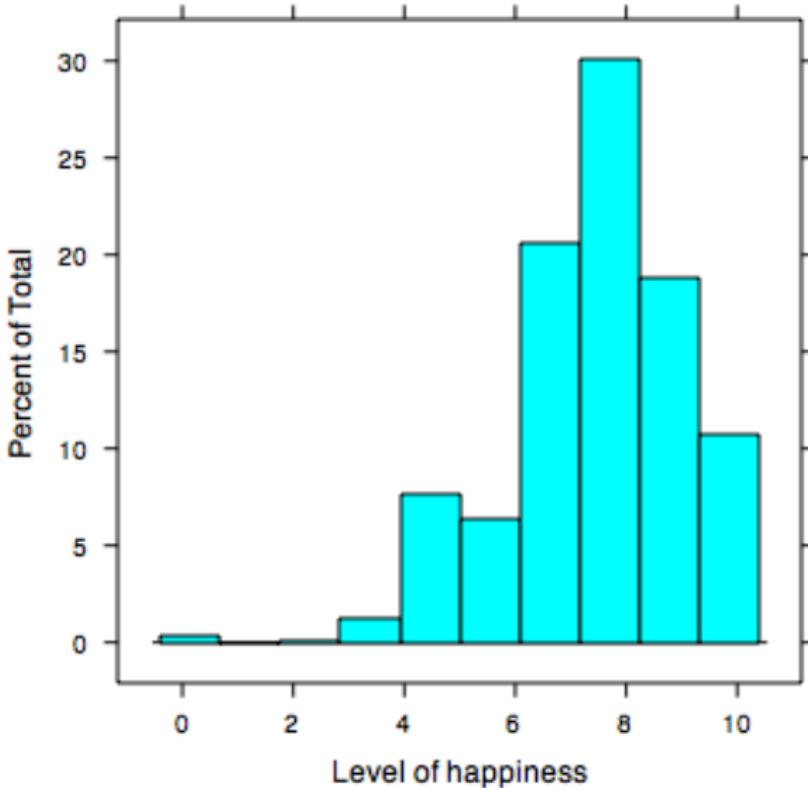
- ① 对极大整数做因数分解非常困难（破解 RSA-2048（2048-bit）的密钥可能需要耗费传统电脑 10 亿年的时间，而量子计算机只需要 100 秒就可以完成。）【量子计算机 + 秀尔算法】
- ② 普通台式机，每秒运算 40 亿次，需要大约 30 万年以上才能破解
- ③ 利文索尔佯谬（Levinthal's paradox）：100 氨基酸，每个 2 种构象，每次尝试耗时  $10^{-13}$ s，穷举需要 40 亿年

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

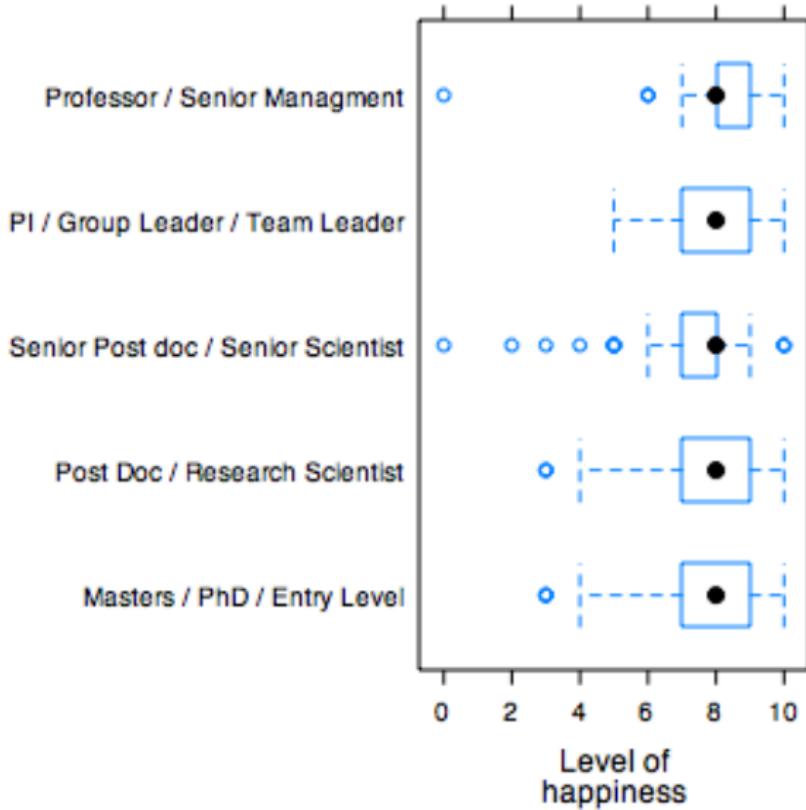
- 6 Bioinformatics Career Survey  
2008
- 7 R
- 8 回顾与总结
  - 总结
  - 思考题



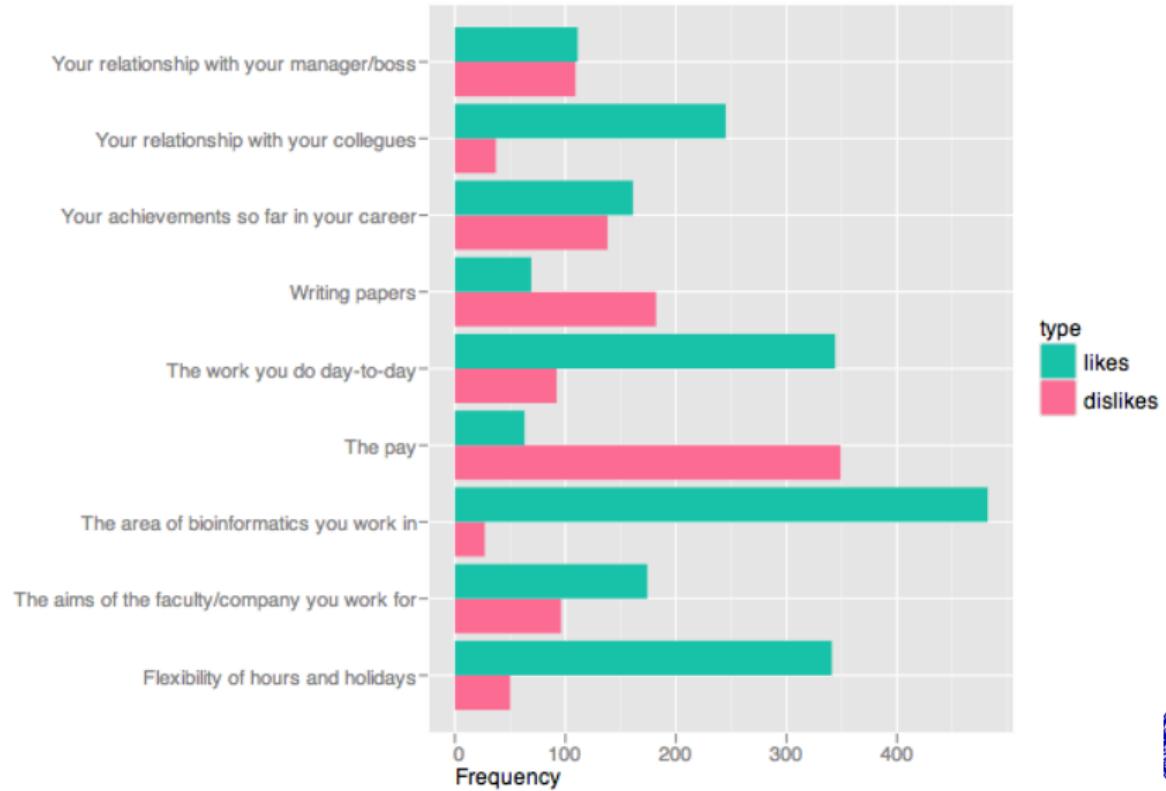
# 2008 | happiness

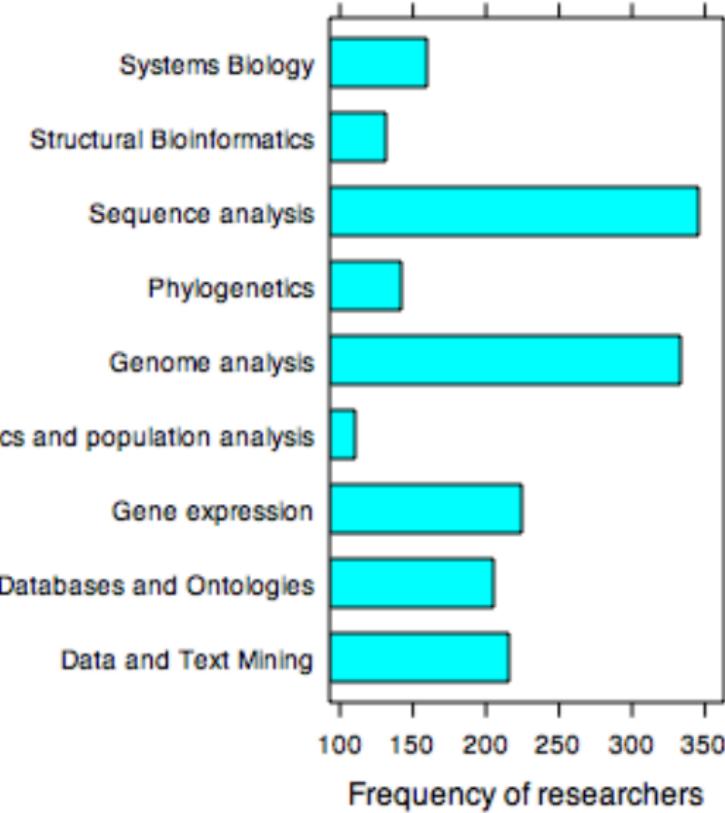


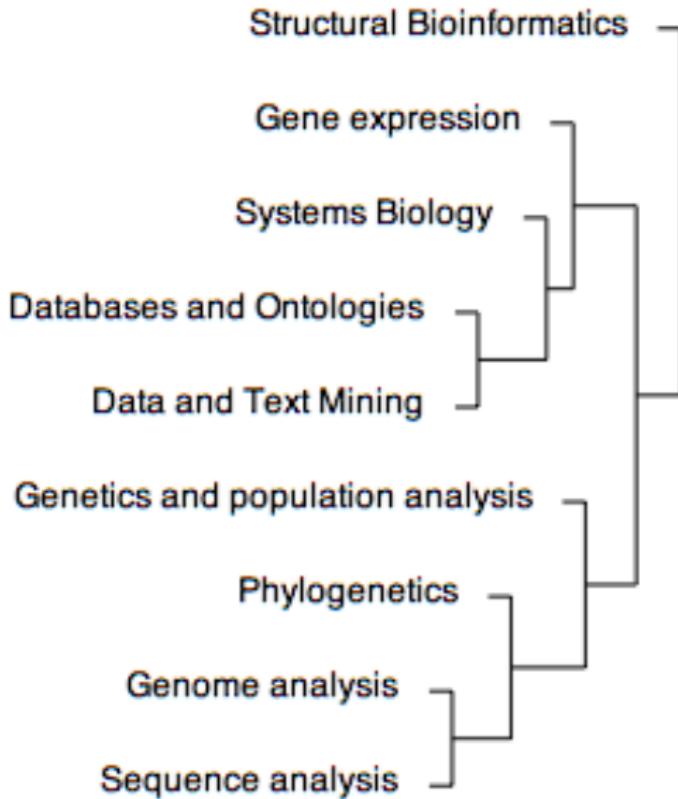
# 2008 | happiness vs. career position



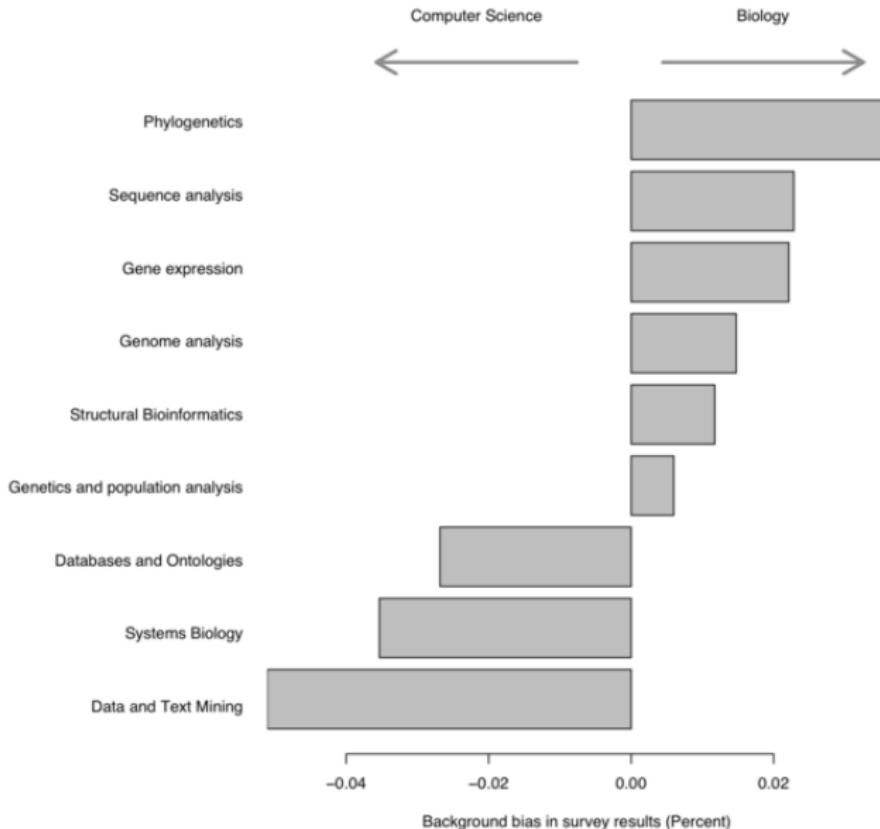
# 2008 | likes vs. dislikes



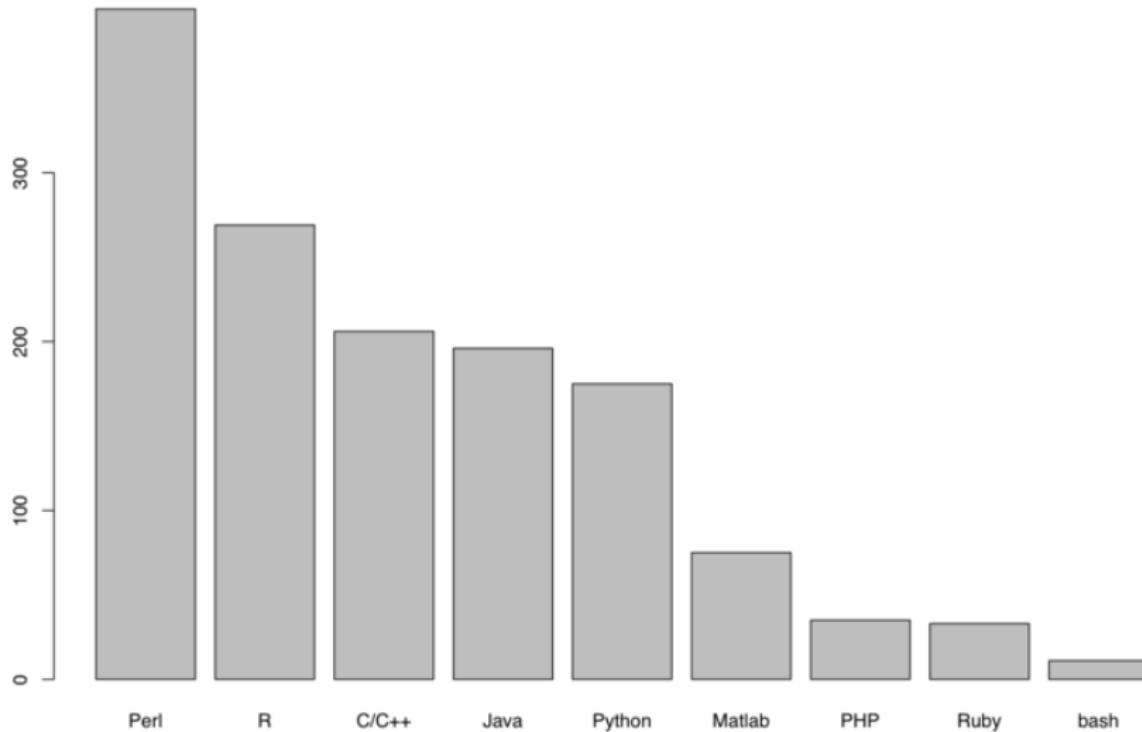




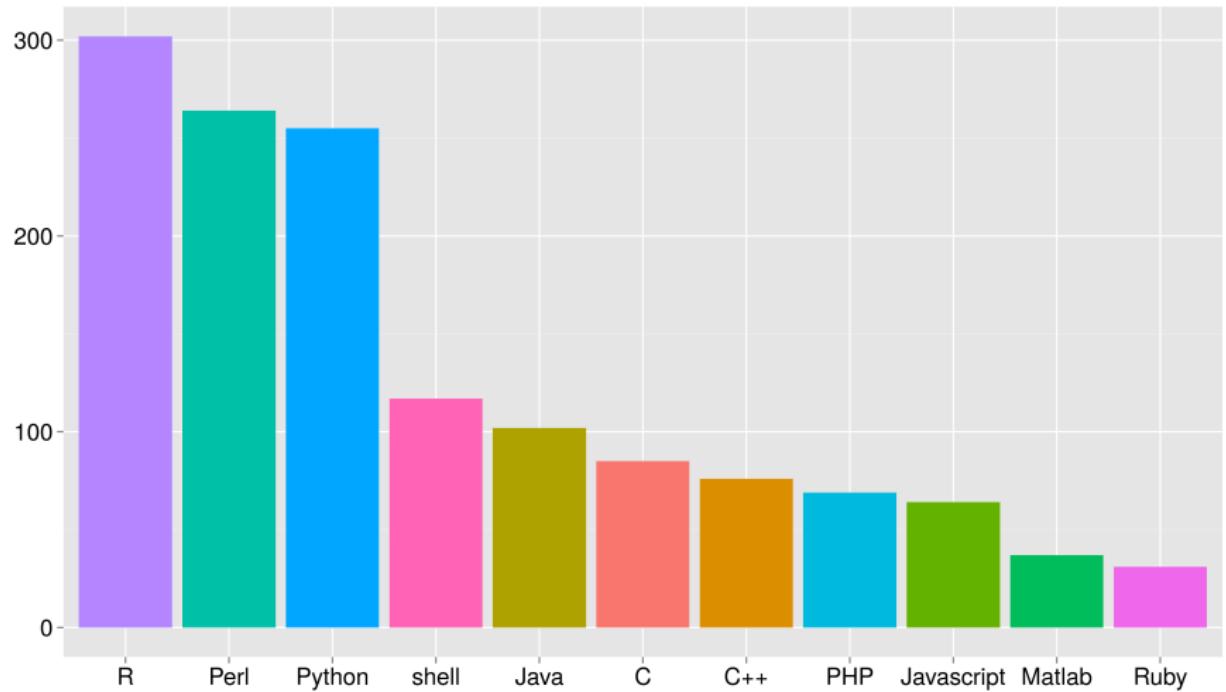
# 2008 | background



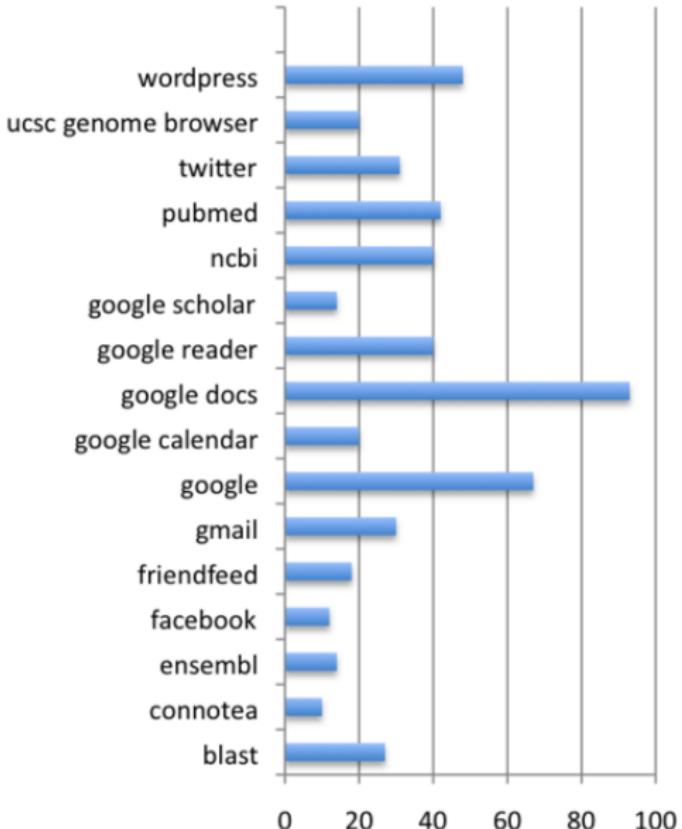
# 2008 | language

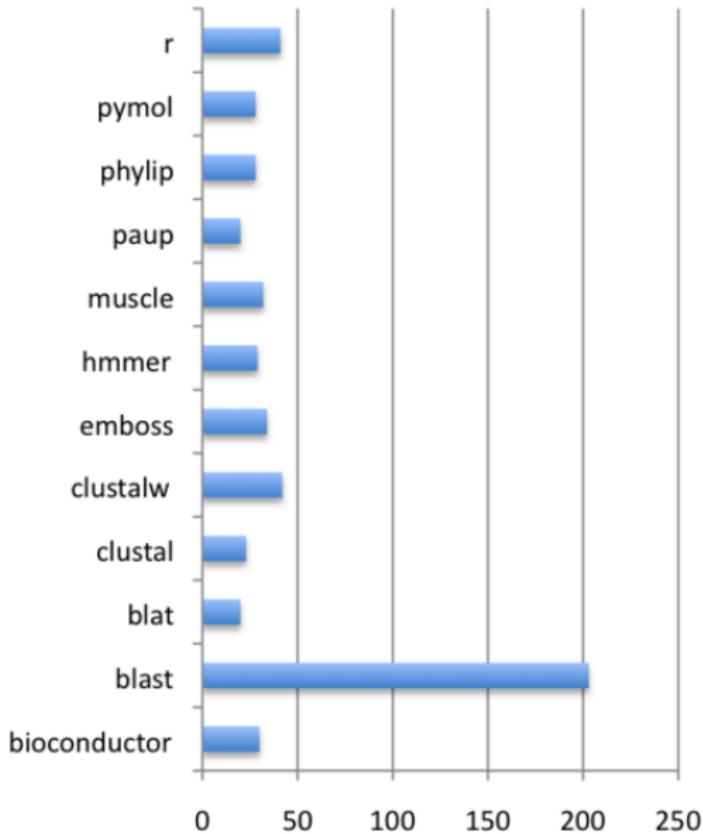


# 2012 | language



# 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

- 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 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.

## 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).

## 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.

## 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).

## 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.

**XLConnect** Help you read, write and format **Microsoft 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.



## 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.



## 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.



- 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 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 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



- 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 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。



## 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<sub>E</sub>X L<sup>A</sup>T<sub>E</sub>X X<sub>E</sub>T<sub>E</sub>X Beamer