

出芽酵母单核苷酸突变特征

成员与分工

- 生信 2001 张子栋
 - 代码实现 (Java 代码)
 - 项目展示
 - 实验报告书
- 生信 2001 梁国相
 - 思路与模型建立
 - 代码校对
 - R 语言绘图

基因序列和非基因序列突变率的差异

实验思路

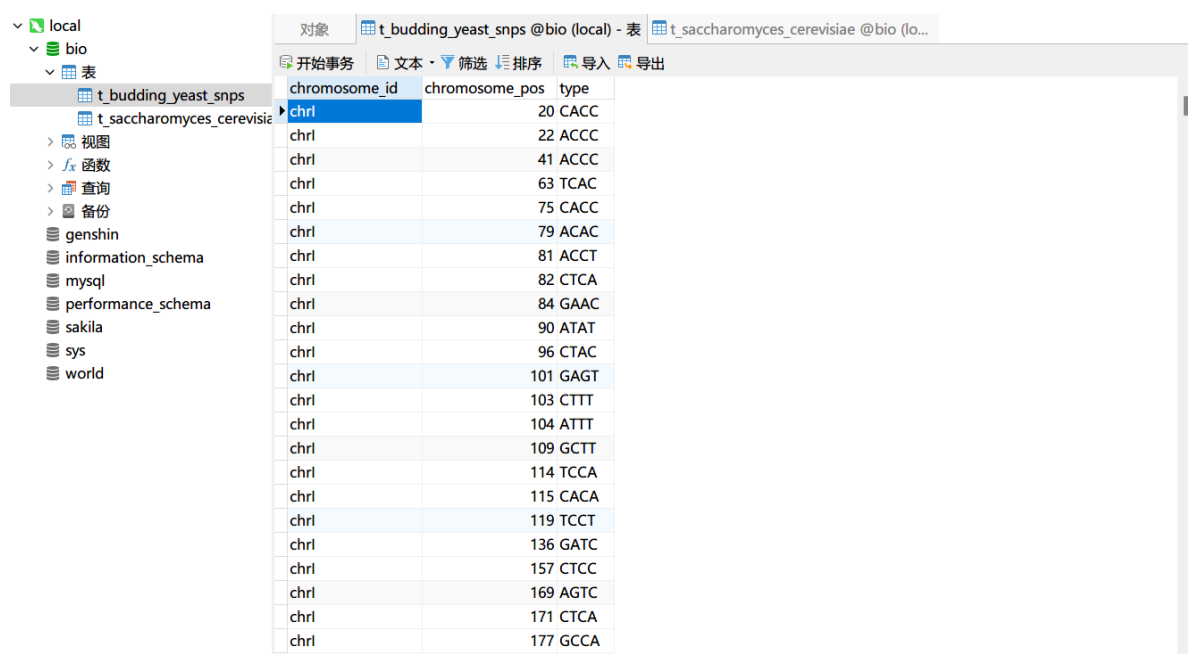
使用 Java 通过 MyBatis 操作数据库，完成统计量计算；再通过 R 绘图。

数据预处理

数据导入

利用 Navicat 导入向导，导入 Budding Yeast_SNPs.txt saccharomyces_cerevisiae_R64-1-1_20110208.gff 文件到 MySQL 数据库

其中，saccharomyces_cerevisiae_R64-1-1_20110208.gff 文件仅提取第 1, 3, 4, 5 列为有效信息



对象	t_budding_yeast_snps @bio (local) - 表	t_saccharomyces_cerevisiae @bio (lo...
开始事务	文本	筛选
排序	导入	导出
chromosome_id	chromosome_pos	type
chr1	20	CACC
chr1	22	ACCC
chr1	41	ACCC
chr1	63	TCAC
chr1	75	CACC
chr1	79	ACAC
chr1	81	ACCT
chr1	82	CTCA
chr1	84	GAAC
chr1	90	ATAT
chr1	96	CTAC
chr1	101	GAGT
chr1	103	CTTT
chr1	104	ATTT
chr1	109	GCTT
chr1	114	TCCA
chr1	115	CACA
chr1	119	TCCT
chr1	136	GATC
chr1	157	CTCC
chr1	169	AGTC
chr1	171	CTCA
chr1	177	GCCA

local

bio

表

t_budding_yeast_snps

t_saccharomyces_cerevisiae

视图

函数

查询

备份

genshin

information_schema

mysql

performance_schema

sakila

sys

world

对象

t_budding_yeast_snps @bio (local) - 表

t_saccharomyces_cerevisiae @bio (lo...

开始事务

文本

筛选

排序

导入

导出

chromosome_id	type	start_pos	end_pos
chr1	chromosome	1	230218
chr1	repeat_region	1	62
chr1	telomere	1	801
chr1	repeat_region	63	336
chr1	gene	335	649
chr1	CDS	335	649
chr1	repeat_region	337	801
chr1	nucleotide_match	753	763
chr1	binding_site	532	544
chr1	gene	538	792
chr1	CDS	538	792
chr1	ARS	650	1791
chr1	gene	1807	2169
chr1	CDS	1807	2169
chr1	gene	2480	2707
chr1	CDS	2480	2707
chr1	gene	7235	9016
chr1	CDS	7235	9016
chr1	ARS	7997	8547
chr1	gene	10091	10399
chr1	CDS	10091	10399
chr1	gene	11565	11951
chr1	CDS	11565	11951

创建实体类

Information 类, 存储基因位置

```

1 package com.zidongzh.pojo;
2
3 /**
4  * @author zidongzh
5  * @date 2022/3/3
6  */
7 public class Information {
8     private String chromosomeId;
9     private String type;
10    private Integer startPos;
11    private Integer endPos;
12
13    /**
14     * constructor
15     *
16     * @param chromosomeId 染色体号
17     * @param type          类型 (基因, 非基因)
18     * @param startPos      起始位置
19     * @param endPos        终止位置
20     */
21    public Information(
22        String chromosomeId,
23        String type,
24        Integer startPos,
25        Integer endPos) {
26        this.chromosomeId = chromosomeId;
27        this.type = type;
28        this.startPos = startPos;
29        this.endPos = endPos;
30    }
31
32    @Override
33    public String toString() {
34        return "Information{" +

```

```

35         "chromosomeId='" + chromosomeId + '\'' +
36         ", type='" + type + '\'' +
37         ", startPos=" + startPos +
38         ", endPos=" + endPos +
39         '}'';
40     }
41
42     // getter and setter
43
44
45     public String getChromosomeId() {
46         return chromosomeId;
47     }
48
49     public void setChromosomeId(String chromosomeId) {
50         this.chromosomeId = chromosomeId;
51     }
52
53     public String getType() {
54         return type;
55     }
56
57     public void setType(String type) {
58         this.type = type;
59     }
60
61     public Integer getStartPos() {
62         return startPos;
63     }
64
65     public void setStartPos(Integer startPos) {
66         this.startPos = startPos;
67     }
68
69     public Integer getEndPos() {
70         return endPos;
71     }
72
73     public void setEndPos(Integer endPos) {
74         this.endPos = endPos;
75     }
76
77 }
78

```

Mutation 类, 存储变异信息。

```

1  package com.zidongzh.pojo;
2
3  /**
4   * @author ZidongZh
5   * @date 2022/3/3
6   */
7  public class Mutation {
8      private String chromosomeId;
9      private Integer chromosomePos;
10     private String type;

```

```

11
12     /**
13      * constructor
14      *
15      * @param chromosomeId 染色体号
16      * @param chromosomePos 染色体位置
17      * @param type 变异类型
18      */
19     public Mutation(
20         String chromosomeId,
21         Integer chromosomePos,
22         String type) {
23         this.chromosomeId = chromosomeId;
24         this.chromosomePos = chromosomePos;
25         this.type = type;
26     }
27
28     @Override
29     public String toString() {
30         return "Mutation{" +
31             "chromosomeId='" + chromosomeId + '\'' +
32             ", chromosomePos=" + chromosomePos +
33             ", type='" + type + '\'' +
34             '}';
35     }
36
37     // getter and setter
38
39     public String getChromosomeId() {
40         return chromosomeId;
41     }
42
43     public void setChromosomeId(String chromosomeId) {
44         this.chromosomeId = chromosomeId;
45     }
46
47     public Integer getChromosomePos() {
48         return chromosomePos;
49     }
50
51     public void setChromosomePos(Integer chromosomePos) {
52         this.chromosomePos = chromosomePos;
53     }
54
55     public String getType() {
56         return type;
57     }
58
59     public void setType(String type) {
60         this.type = type;
61     }
62 }
63

```

数据处理

对 `saccharomyces_cerevisiae_R64-1-1_20110208` 数据中基因重叠起始位置的处理

```

1  /**
2   * 合并重叠基因区间
3   *
4   * @param information 原始数据
5   */
6  void connect(List<Information> information) {
7      int i = 0;
8      int j = 1;
9      Information previous = null;
10     Information next = null;
11     while (j < information.size()) {
12         previous = information.get(i);
13         next = information.get(j);
14         if (previous.getEndPos() >= next.getStartPos() &&
previous.getEndPos() <= next.getEndPos()) {
15             previous.setEndPos(next.getEndPos());
16             information.remove(j);
17             information.set(i, previous);
18         } else {
19             i++;
20             j++;
21         }
22     }
23 }

```

获取非基因区间

```

1  /**
2   * @param genes 基因信息
3   * @param chromosomes 染色体信息
4   * @return 非基因区间
5   */
6  List<Information> getNonGenes(List<Information> genes,
7                               List<Information> chromosomes) {
8      int pos = 0;
9
10     List<Information> nonGenes = new ArrayList<>();
11     Information nonGeneHead = new
Information(chromosomes.get(0).getChromosomeId(), "nonGene", 1,
genes.get(0).getStartPos() - 1);
12     nonGenes.add(nonGeneHead);
13     for (int i = 0; i < chromosomes.size(); i++) {
14         for (int j = 0; j < genes.size(); j++) {
15             if
(genes.get(j).getChromosomeId().equals(chromosomes.get(i).getChromosomeId())
&&
16                 genes.get(j +
1).getChromosomeId().equals(chromosomes.get(i).getChromosomeId())) {
17                 Information nonGene = new
Information(genes.get(j).getChromosomeId(), "nonGene",
genes.get(j).getEndPos() + 1, genes.get(j + 1).getStartPos() - 1);
18                 nonGenes.add(nonGene);
19                 pos = j + 2;
20             }
21         }
22         if (null != genes.get(pos)){

```

```
23         System.out.println(genes.get(pos));
24         Information nonGeneHead1 = new
Information(genes.get(pos).getChromosomeId(), "nonGene", 1,
genes.get(pos).getStartPos() - 1);
25         nonGenes.add(nonGeneHead1);
26     }
27 }
28 return nonGenes;
29 }
```

统计变异数

在 `Information` 类中添加属性 `mutationNum` 与 `mutationRate`

```

public class Information {
    private String chromosomeId;
    private String type;
    private Integer startPos;
    private Integer endPos;
    private Integer mutationNum;
    private Double mutationRate;

    /**
     * constructor
     *
     * @param chromosomeId 染色体号
     * @param type          类型（基因，非基因）
     * @param startPos      起始位置
     * @param endPos        终止位置
     * @param mutationNum   变异数
     */
    public Information(
        String chromosomeId,
        String type,
        Integer startPos,
        Integer endPos,
        Integer mutationNum,
        Double mutationRate ) {
        this.chromosomeId = chromosomeId;
        this.type = type;
        this.startPos = startPos;
        this.endPos = endPos;
        this.mutationNum = mutationNum;
        this.mutationRate = mutationRate;
    }
}

```

获取变异数 计算变异率 并存入数据库

```

1 //计算变异数与变异率
2 for (int i = 0; i < allMutation.size(); i++) {
3     for (int j = 0; j < genes.size(); j++) {
4         if
5             (allMutation.get(i).getChromosomeId().equals(genes.get(j).getChromosomeId())
6             && allMutation.get(i).getChromosomePos() >= genes.get(j).getStartPos() &&
7             allMutation.get(i).getChromosomePos() <=
8             genes.get(j).getEndPos()) {
9                 genes.get(j).setMutationNum(genes.get(j).getMutationNum() + 1);

```

```

7         genes.get(j).setMutationRate((double)
genes.get(j).getMutationNum() / (double) (genes.get(j).getEndPos() -
genes.get(j).getStartPos() + 1));
8     }
9 }
10 for (int j = 0; j < nonGenes.size(); j++) {
11     if
(allMutation.get(i).getChromosomeId().equals(nonGenes.get(j).getChromosomeId
()) && allMutation.get(i).getChromosomePos() >=
nonGenes.get(j).getStartPos() &&
12         allMutation.get(i).getChromosomePos() <=
nonGenes.get(j).getEndPos()) {
13         nonGenes.get(j).setMutationNum(nonGenes.get(j).getMutationNum()
+ 1);
14         nonGenes.get(j).setMutationRate((double)
nonGenes.get(j).getMutationNum() / (double) (nonGenes.get(j).getEndPos() -
nonGenes.get(j).getStartPos() + 1));
15     }
16 }
17 }
18
19 //将基因与非基因添加到数据库
20 for (int i = 0; i < genes.size(); i++) {
21     informationMapper.addGeneInfo(genes.get(i));
22 }
23 for (int i = 0; i < nonGenes.size(); i++) {
24     informationMapper.addNonGeneInfo(nonGenes.get(i));
25 }
26
27 //提交修改至数据库
28 sqlSession.commit();

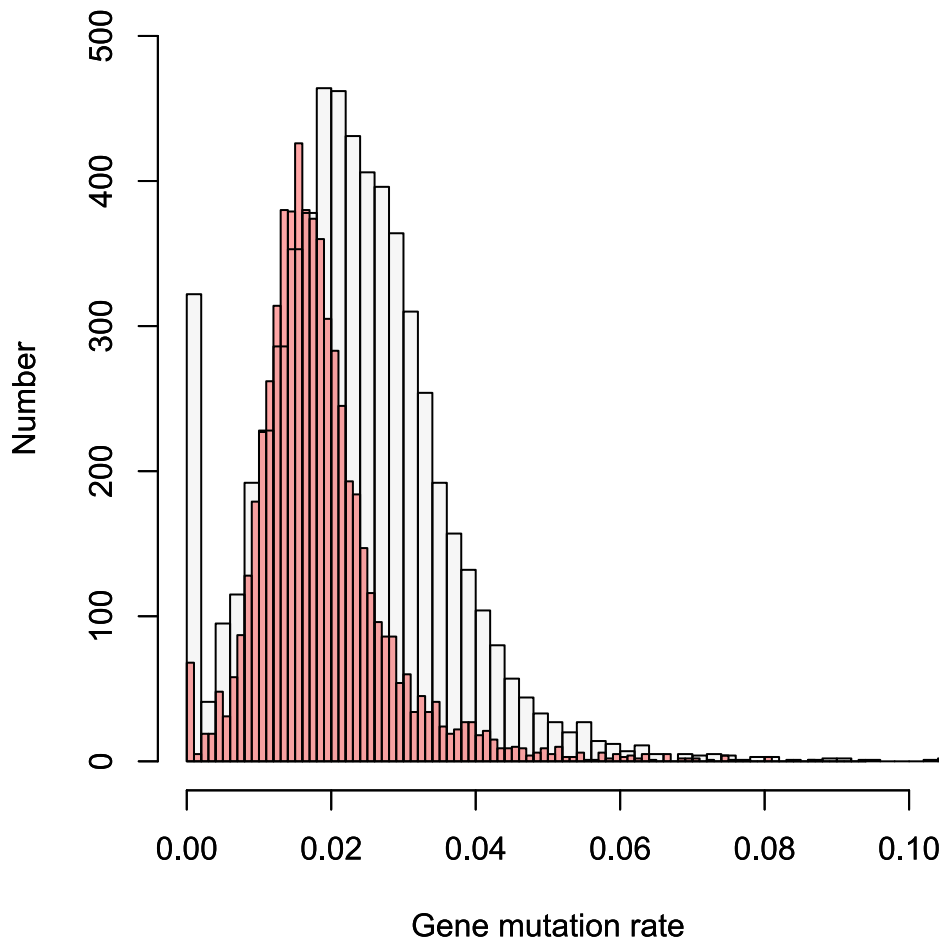
```

数据可视化

```

1 > geneMutationRate = read.table(file.choose(), header = F, sep = ",")
2 > geneMutationRate = as.matrix(geneMutationRate)
3 > nonGeneMutationRate = read.table(file.choose(), header = F, sep = ",")
4 > nonGeneMutationRate = as.matrix(nonGeneMutationRate)
5 > hist(geneMutationRate, xlab="Gene mutation rate",ylab="Number",col=rgb(255,
0, 0, 90, maxColorValue=255), breaks = 100, xlim = c(0.0,0.1), ylim =
c(0,500), main = NULL)
6 > par(new = T)
7 > hist(nonGeneMutationRate, xlab="Gene mutation
rate",ylab="Number",col=rgb(2, 0, 0, 10, maxColorValue=255), breaks = 100,
xlim = c(0.0,0.1), ylim = c(0,500), main = NULL)

```

假设检验

```
1      @Test
2      public void bioTest() throws IOException {
3          InputStream inputStream = Resources.getResourceAsStream("mybatis-
4      config.xml");
5          SqlSessionFactoryBuilder sqlSessionFactoryBuilder = new
6      SqlSessionFactoryBuilder();
7          SqlSessionFactory sqlSessionFactory =
8      sqlSessionFactoryBuilder.build(inputStream);
9          SqlSession sqlSession = sqlSession.openSession();
10
11         InformationMapper informationMapper =
12         sqlSession.getMapper(InformationMapper.class);
13
14         List<Information> genes = informationMapper.getGenes();
15         List<Information> nonGenes = informationMapper.getNonGenes();
16
17         double geneMutRateMean = 0.0;
18         double geneVariance = 0.0;
```

```
16         double nonGeneMutRateMean = 0.0;
17         double nonGeneVariance = 0.0;
```

```

19     geneMutRateMean = getMean(genes);
20     geneVariance = getVariance(genes, geneMutRateMean);
21     System.out.println("geneMutRateMean = " + geneMutRateMean);
22     System.out.println("geneVariance = " + geneVariance);
23
24     nonGeneMutRateMean = getMean(nonGenes);
25     nonGeneVariance = getVariance(nonGenes, nonGeneMutRateMean);
26     System.out.println("nonGeneMutRateMean = " + nonGeneMutRateMean);
27     System.out.println("nonGeneVariance = " + nonGeneVariance);
28
29     //t检验
30     double t = 0.0;
31     t = (geneMutRateMean - nonGeneMutRateMean) / (sqrt((((genes.size() -
1) * geneVariance) + ((nonGenes.size() - 1) * nonGeneVariance)) /
(genes.size() + nonGenes.size() - 2)) * sqrt((1 / (double) genes.size()) +
(1 / (double) nonGenes.size())));
32     System.out.println("t = " + t);
33     //u 检验
34     double u = 0.0;
35     u = (geneMutRateMean - nonGeneMutRateMean) / (sqrt((geneVariance /
genes.size()) + (nonGeneVariance / nonGenes.size())));
36     System.out.println("u = " + u);
37 }

```

运行结果:

```

1 geneMutRateMean = 0.018604969719609406
2 geneVariance = 8.597880264141603E-5
3 nonGeneMutRateMean = 0.023312667924136225
4 nonGeneVariance = 1.7675824048238737E-4
5 t = -22.612916116843113
6 u = -22.610982585720944

```

U 检验

1. 提出:

- $H_0: \mu_1 = \mu_2$, 出芽酵母基因区间突变率与非基因区间突变率没有显著差异
- $H_A: \mu_1 \neq \mu_2$

2. 假定 H_0 成立

3. 选取显著水平 $\alpha = 0.01$

4. 统计量:

$$s_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

$$u = \frac{(\bar{x}_1 - \bar{x}_2)}{s_{\bar{x}_1 - \bar{x}_2}}$$

设计程序计算得出 $u = -22.610982585720944$

$$|u| > 2.58, P < 0.01$$

5. 推断: 在 0.01 显著水平上, 拒绝 H_0 , 接受 H_A

认为出芽酵母基因区间突变率与非基因区间突变率有显著差异

T 检验

1. 提出:

- $H_0: \mu_1 = \mu_2$, 出芽酵母基因区间突变率与非基因区间突变率没有显著差异
- $H_A: \mu_1 \neq \mu_2$

2. 假定 H_0 成立

3. 选取显著水平 $\alpha = 0.01$

4. 统计量:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

设计程序计算得出 $t = -22.612916116843113$

$$|t| > 2.576, P < 0.01$$

5. 推断: 在 0.01 显著水平上, 拒绝 H_0 , 接受 H_A

认为出芽酵母基因区间突变率与非基因区间突变率有显著差异

使用 R 进行 T 检验

使用 R 内置的 T 检验, 验算结果

```
1 > t.test(geneMutationRate, nonGeneMutationRate)
2
3      welch Two Sample t-test
4
5 data:  geneMutationRate and nonGeneMutationRate
6 t = -22.611, df = 10825, p-value < 2.2e-16
7 alternative hypothesis: true difference in means is not equal to 0
8 95 percent confidence interval:
9  -0.005115816 -0.004299580
10 sample estimates:
11  mean of x  mean of y
12  0.01860497 0.02331267
```

与 Java 计算出的结果一致

突变方向的差异性

修改实体类

```

/**
 * @author ZidongZh
 * @date 2022/3/3
 */
public class Information {
    private String chromosomeId;
    private String type;
    private Integer startPos;
    private Integer endPos;
    private Integer mutationNum;
    private Double mutationRate;
    private Integer ANum;
    private Integer CNum;
    private Integer GNum;
    private Integer TNum;
    private Integer ACNum;
    private Integer AGNum;
    private Integer TCNum;
    private Integer TGNum;
    private Integer CTNum;
    private Integer CANum;
    private Integer GTNum;
    private Integer GCNum;
    private Integer ATNum;
}

```

算法设计

将基因的突变方向及其突变数量存储到数据库中

```

1 void classify(Information gene, Mutation mutation) {
2     if (mutation.getChromosomeId().equals(gene.getChromosomeId()) &&
3         mutation.getChromosomePos() >= gene.getStartPos() &&
4         mutation.getChromosomePos() <= gene.getEndPos()) {
5         if (mutation.getType().startsWith("AT")) {
6             gene.setATNum(gene.getATNum() + 1);
7         } else if (mutation.getType().startsWith("AG")) {
8             gene.setAGNum(gene.getAGNum() + 1);
9         } else if (mutation.getType().startsWith("AC")) {
10            gene.setACNum(gene.getACNum() + 1);
11        } else if (mutation.getType().startsWith("TA")) {
12            gene.setTANum(gene.getTANum() + 1);
13        } else if (mutation.getType().startsWith("TG")) {
14            gene.setTGNum(gene.getTGNum() + 1);
15        } else if (mutation.getType().startsWith("TC")) {

```

```

16         gene.setTCNum(gene.getTCNum() + 1);
17     } else if (mutation.getType().startsWith("GA")) {
18         gene.setGNum(gene.getGNum() + 1);
19     } else if (mutation.getType().startsWith("GT")) {
20         gene.setGTNum(gene.getGTNum() + 1);
21     } else if (mutation.getType().startsWith("GC")) {
22         gene.setGCNum(gene.getGCNum() + 1);
23     } else if (mutation.getType().startsWith("CA")) {
24         gene.setCNum(gene.getCNum() + 1);
25     } else if (mutation.getType().startsWith("CT")) {
26         gene.setCTNum(gene.getCTNum() + 1);
27     } else if (mutation.getType().startsWith("CG")) {
28         gene.setCGNum(gene.getCGNum() + 1);
29     }
30 }
31 }

```

t 检验算法

```

1  /**
2   * t检验
3   *
4   * @param genes    基因序列数据
5   * @param nonGenes 非基因序列数据
6   * @param flag      需要计算的突变类型 "AG" 表示由 A 突变为 G
7   * @return
8   */
9  double tTest(List<Information> genes, List<Information> nonGenes, String
flag) {
10
11      List<Double> geneRate = rate(genes, flag);
12      List<Double> nonGeneRate = rate(nonGenes, flag);
13      double geneMutRateMean = 0.0;
14      double nonGeneMutRateMean = 0.0;
15      double geneSum = 0.0;
16      double nonGeneSum = 0.0;
17      double geneVariance = 0.0;
18      double nonGeneVariance = 0.0;
19      double t = 0.0;
20
21      geneMutRateMean = mean(geneRate);
22      nonGeneMutRateMean = mean(nonGeneRate);
23      geneSum = (double) genes.size();
24      nonGeneSum = (double) nonGenes.size();
25      geneVariance = variance(geneRate, geneMutRateMean);
26      nonGeneVariance = variance(nonGeneRate, nonGeneMutRateMean);
27
28      t = (geneMutRateMean - nonGeneMutRateMean) / (sqrt((((geneSum - 1) *
geneVariance) + ((nonGeneSum - 1) * nonGeneVariance)) / (geneSum +
nonGeneSum - 2))) * sqrt((1 / geneSum) + (1 / nonGeneSum)));
29      System.out.println("gene mean is " + geneMutRateMean);
30      System.out.println("non gene mean is " + nonGeneMutRateMean);
31      return t;
32  }

```

u 检验算法

```
1  /**
2   * u检验
3   *
4   * @param genes      基因序列数据
5   * @param nonGenes   非基因序列数据
6   * @param flag       需要计算的突变类型 “AG” 表示由 A 突变为 G
7   * @return
8   */
9  double uTest(List<Information> genes, List<Information> nonGenes, String
flag) {
10     List<Double> geneRate = rate(genes, flag);
11     List<Double> nonGeneRate = rate(nonGenes, flag);
12     double geneMutRateMean = 0.0;
13     double nonGeneMutRateMean = 0.0;
14     double geneSum = 0.0;
15     double nonGeneSum = 0.0;
16     double geneVariance = 0.0;
17     double nonGeneVariance = 0.0;
18     double u = 0.0;
19
20     geneMutRateMean = mean(geneRate);
21     nonGeneMutRateMean = mean(nonGeneRate);
22     geneSum = (double) genes.size();
23     nonGeneSum = (double) nonGenes.size();
24     geneVariance = variance(geneRate, geneMutRateMean);
25     nonGeneVariance = variance(nonGeneRate, nonGeneMutRateMean);
26
27     u = (geneMutRateMean - nonGeneMutRateMean) / (sqrt((geneVariance /
geneSum) + (nonGeneVariance / nonGeneSum)));
28     return u;
29 }
```

辅助算法

在 t 检验和 u 检验中调用的方法:

```
1  /**
2   * 由原始数据计算突变率
3   *
4   * @param information 输入的序列
5   * @param flag       需要计算的突变类型 “AG” 表示由 A 突变为 G
6   * @return 返回突变率集合
7   */
8  List<Double> rate(List<Information> information, String flag) {
9     int num = 0;
10     int length = 0;
11     List<Double> rates = new ArrayList<>();
12     for (int i = 0; i < information.size(); i++) {
13         length = information.get(i).getEndPos() -
information.get(i).getStartPos() + 1;
14         if ("AT".equals(flag)) {
15             num = information.get(i).getATNum();
16         } else if ("AG".equals(flag)) {
17             num = information.get(i).getAGNum();
18         }
19     }
20     return rates;
21 }
```

```

18         } else if ("AC".equals(flag)) {
19             num = information.get(i).getACNum();
20         } else if ("TA".equals(flag)) {
21             num = information.get(i).getTANum();
22         } else if ("TG".equals(flag)) {
23             num = information.get(i).getTGNum();
24         } else if ("TC".equals(flag)) {
25             num = information.get(i).getTCNum();
26         } else if ("GA".equals(flag)) {
27             num = information.get(i).getGANum();
28         } else if ("GT".equals(flag)) {
29             num = information.get(i).getGTNum();
30         } else if ("GC".equals(flag)) {
31             num = information.get(i).getGCNum();
32         } else if ("CA".equals(flag)) {
33             num = information.get(i).getACNum();
34         } else if ("CT".equals(flag)) {
35             num = information.get(i).getCTNum();
36         } else if ("CG".equals(flag)) {
37             num = information.get(i).getCGNum();
38         }
39         rates.add((double) num / (double) length);
40     }
41     return rates;
42 }

```

```

1  /**
2   * 计算突变率均值
3   *
4   * @param rates 突变率集合
5   * @return 返回均值
6   */
7  double mean(List<Double> rates) {
8      double sum = 0.0;
9      for (int i = 0; i < rates.size(); i++) {
10         //Double是引用数据类型 此处 if 语句用于防止自动装箱时产生误差
11         if (rates.get(i) > 0.0) {
12             sum += rates.get(i);
13         }
14     }
15     return sum / rates.size();
16 }

```

```

1  /**
2   * 计算方差
3   *
4   * @param rates 突变率集合
5   * @param mean 突变率均值
6   * @return 返回方差
7   */
8  double variance(List<Double> rates, double mean) {
9      double sum = 0.0;
10     for (int i = 0; i < rates.size(); i++) {
11         //Double是引用数据类型 此处 if 语句用于防止自动装箱时产生误差
12         if (rates.get(i) > 0.0) {
13             sum += pow((rates.get(i) - mean), (double) 2);

```

```

14         }
15     }
16     return sum / (rates.size() - 1);
17 }

```

假设检验

代码实现

在测试类中编写代码 计算统计量的值

```

1  @Test
2      public void mutationTest() throws IOException {
3      InputStream inputStream = Resources.getResourceAsStream("mybatis-
config.xml");
4      SqlSessionFactoryBuilder sqlSessionFactoryBuilder = new
SqlSessionFactoryBuilder();
5      SqlSessionFactory sqlSessionFactory =
sqlSessionFactoryBuilder.build(inputStream);
6      SqlSession sqlSession = sqlSessionFactory.openSession();
7
8      InformationMapper informationMapper =
sqlSession.getMapper(InformationMapper.class);
9      MutationMapper mutationMapper =
sqlSession.getMapper(MutationMapper.class);
10
11      List<Information> genes = informationMapper.getGenes();
12      List<Information> nonGenes = informationMapper.getNonGenes();
13
14      System.out.println("-----");
15      System.out.println("AC: t is " + tTest(genes, nonGenes, "AC"));
16      System.out.println("-----");
17      System.out.println("AT: t is " + tTest(genes, nonGenes, "AT"));
18      System.out.println("-----");
19      System.out.println("GA: t is " + tTest(genes, nonGenes, "GA"));
20      System.out.println("-----");
21      System.out.println("GC: t is " + tTest(genes, nonGenes, "GC"));
22      System.out.println("-----");
23      System.out.println("GT: t is " + tTest(genes, nonGenes, "GT"));
24      System.out.println("-----");
25      System.out.println("CA: t is " + tTest(genes, nonGenes, "CA"));
26      System.out.println("-----");
27      System.out.println("CG: t is " + tTest(genes, nonGenes, "CG"));
28      System.out.println("-----");
29      System.out.println("CT: t is " + tTest(genes, nonGenes, "CT"));
30      System.out.println("-----");
31      System.out.println("TA: t is " + tTest(genes, nonGenes, "TA"));
32      System.out.println();
33      System.out.println();
34      System.out.println("-----");
35      System.out.println("AC: u is " + uTest(genes, nonGenes, "AC"));
36      System.out.println("-----");
37      System.out.println("AT: u is " + uTest(genes, nonGenes, "AT"));
38      System.out.println("-----");
39      System.out.println("GA: u is " + uTest(genes, nonGenes, "GA"));
40      System.out.println("-----");

```



```

41     System.out.println("GC: u is " + uTest(genes, nonGenes, "GC"));
42     System.out.println("-----");
43     System.out.println("GT: u is " + uTest(genes, nonGenes, "GT"));
44     System.out.println("-----");
45     System.out.println("CA: u is " + uTest(genes, nonGenes, "CA"));
46     System.out.println("-----");
47     System.out.println("CG: u is " + uTest(genes, nonGenes, "CG"));
48     System.out.println("-----");
49     System.out.println("CT: u is " + uTest(genes, nonGenes, "CT"));
50     System.out.println("-----");
51     System.out.println("TA: u is " + uTest(genes, nonGenes, "TA"));
52 }

```

输出结果

```

1  -----
2  AC: t is -12.737652036387919
3  -----
4  AT: t is -21.419035715474898
5  -----
6  GA: t is -2.4649230471397994
7  -----
8  GC: t is -8.191668558555735
9  -----
10 GT: t is -13.803584880399859
11 -----
12 CA: t is -12.737652036387919
13 -----
14 CG: t is -5.413376607132111
15 -----
16 CT: t is -5.337783238152486
17 -----
18 TA: t is -19.556575936619616
19
20
21 -----
22 AC: u is -12.737652036387919
23 -----
24 AT: u is -21.419035715474898
25 -----
26 GA: u is -2.4649230471397994
27 -----
28 GC: u is -8.191668558555735
29 -----
30 GT: u is -13.80358488039986
31 -----
32 CA: u is -12.737652036387919
33 -----
34 CG: u is -5.413376607132111
35 -----
36 CT: u is -5.337783238152486
37 -----
38 TA: u is -19.556575936619616

```

推断方法与基因序列和非基因序列突变率差异相同

生物学意义

- 根据假设检验结果:
 - 基因序列和非基因序列突变率存在显著差异
 - 不同突变方向在基因序列和非基因序列之间存在显著差异
 - 不同突变方向之间部分存在显著差异

完整代码

- 源代码
 - GitHub: [Bluuur/BiostaticsProject1: 生物统计学课程项目1 \(github.com\)](#)
 - 同步至 Gitee: [BiostaticsProject1: 生物统计学课程项目1 \(gitee.com\)](#)
- 实验报告
 - GitHub: [MarkdownNotes/Biostatics at main · Bluuur/MarkdownNotes \(github.com\)](#)
 - 同步至 Gitee: [Biostatics · blur/MarkdownNotes - 码云 - 开源中国 \(gitee.com\)](#)