第六次实验作业

24 MAY 2021

> 实验背景
>
> > 单核苷酸多态性（Single Nucleotide Polymorphism, SNP）指种群中存在于不低于一定比例个体的 DNA 序列内单个核苷酸替代这样的基因多样性[1]。对 SNP 的研究不仅有助于我们研究基因结构，对于基因分型、精准医药方面都有不可忽视的作用[2]；同样也有诸如全基因组关联分析（Genome-Wide Association Study, GWAS）等研究工具利用 SNP 来进行此方面的研究。而对于 SNP 本身而言，又可依照其在基因组上的位置划分为编码区 SNP、非编码区 SNP、间隔区 SNP，因此如何对于 SNP 的功能进行研究，以及如何对于 SNP 进行批量注释成为一个研究的方向。
>
> 实验目的
>
> > 使用 `ANNOVAR` [3] 工具对第三次课程中分析得出的单核苷酸突变进行批量注释、分析其功能。
>
> 运行环境（见下）

```
platform          x86_64-pc-linux-gnu
arch              x86_64
os                linux-gnu
system            x86_64, linux-gnu
R version         4.1.0 (2021-05-18)
nickname          Camp Pontanezen
radian version    0.5.10
python version    3.7.3
xonsh version     0.9.27
```

SNP 批量注释

实验方法

此前得到的 `vcf` 文件中包含了所用样本中单核苷酸突变情况，使用滤除了增删类型的文件，并使用 `ANNOVAR` 进行批量注释。

实验流程

文件格式转换

原始文件 `snp.vcf` 中部分位点发生替换的情况多于一种，因此需要使用 `bcftools`[4] 工具对文件进行一定格式上的转换，将其拆分，并且进行对齐与归一化。

```
bcftools norm -m-both -o out/splited.vcf in/snp.vcf
```

输出结果显示共读到了 1613542 条信息，其中 552 条被拆分。

```
bcftools norm -f in/GRCh38.d1.vd1.fa -o out/normalized.vcf out/splited.vcf
```

输出结果显示共读到 1614094 条信息，无条数变化。

在完成格式上的对齐之后，对于大多数的 `ANNOVAR` 程序而言需要将其转化为统一的 `.avinput` 格式。此处为了方便，后续实验中统一使用 `avinput` 格式。

```
convert2annovar.pl -format vcf4 \
-allsample \
-withfreq \
-includeinfo \
out/normalized.vcf \
> out/prepared.avinput
```

其中使用 `convert2annovar.pl` 转换后的文件主要包含 `vcf` 格式中前五列的内容，`-allsample` 表示将文件中各样品拆分至单独的文件输出（此处仅一个样品）；`-withfreq` 保留等位基因频率；`-includeinfo` 将原文件中所有内容都包含在转换后的文件中。程序的输出见下。

```
NOTICE: Finished reading 1616922 lines from VCF file
NOTICE: A total of 1614094 locus in VCF file passed QC threshold, representing 161374

NOTICE: Finished writing allele frequencies based on 1613748 SNP genotypes (1070337 t

WARNING: 346 invalid alternative alleles found in input file
```

批量注释

参考官方文档的说明，`ANNOVAR` 程序包最简单的使用方法是 `table_annovar.pl` 进行注释，可以“一次性”完成三种类型的注释（基于基因、基于区域和基于过滤的，此处由于缺少基于区域注释所需的数据库文件，因此不进行此项注释）。此函数实际上通过一次性指定数据库与对应操作，相继调用单独的注释程序来完成任务。

```
pkurun-cnlong 1 20 \
    "table_annovar.pl out/prepared.avinput \
software/annovar/humandb/ -buildver hg38 \
-out out/tbl_anno \
-remove \
-protocol refGene,dbnsfp33a \
-operation g,f \
-nastring . \
-csvout"

# -buildver hg38    使用 hg38 版本
# -out out/tbl_anno 指定输出前缀为 out/tbl_anno
# -remove           删除注释过程中的临时文件
# -protocol         指定注释使用的数据库，逗号分隔
# -operation        对应顺序的注释类型（g: gene-based、r: region-based、f: filter-based），

# -nastring .       用点号替代缺省的值
# -csvout           输出格式为 .csv
```

这里我们使用 refGene 数据库来依照 SNP 的位置信息来确定是否为编码序列和 ORF 上非同义突变，进而分析是否影响氨基酸序列。得到的主要注释文件以 `.variant_function` 和 `.exonic_variant_function` 结尾，包含以下内容： 1. `.variant_function` - 包括所有突变的注释 - 第一列为所在基因位置类型 - 第二列为详细描述 - 之后为原始输入的内容 2. `.exonic_variant_function` - 包括外显子非同义突变的注释 - 第一列为该注释在前一文件中出现的行数 - 第二列为该变异的功能性影响 - 第三列为基因名称 - 之后为原始输入的内容

基于过滤的注释用于确认 SNP 是否已记录于数据库，这里使用 LJB (dbNSFP) 全外显子突变数据库来预测其功能。以下为针对参考全外显子突变数据库进行过滤注释的等价写法。由于 LJB 数据库本身较大（~28G），而总过需要注释的条数有 1.6M 条，因此整个注释过程较慢，截至此报告完成，也未得到完整的结果，因此将无法对此部分内容进行后续分析。

```
pkurun-cnlong 1 20 \
    "annotate_variation.pl \
-filter \
-out out/fb_anno \
-build hg38 \
-dbtype dbnsfp33a \
out/prepared.avinput \
software/annovar/humandb/"
```

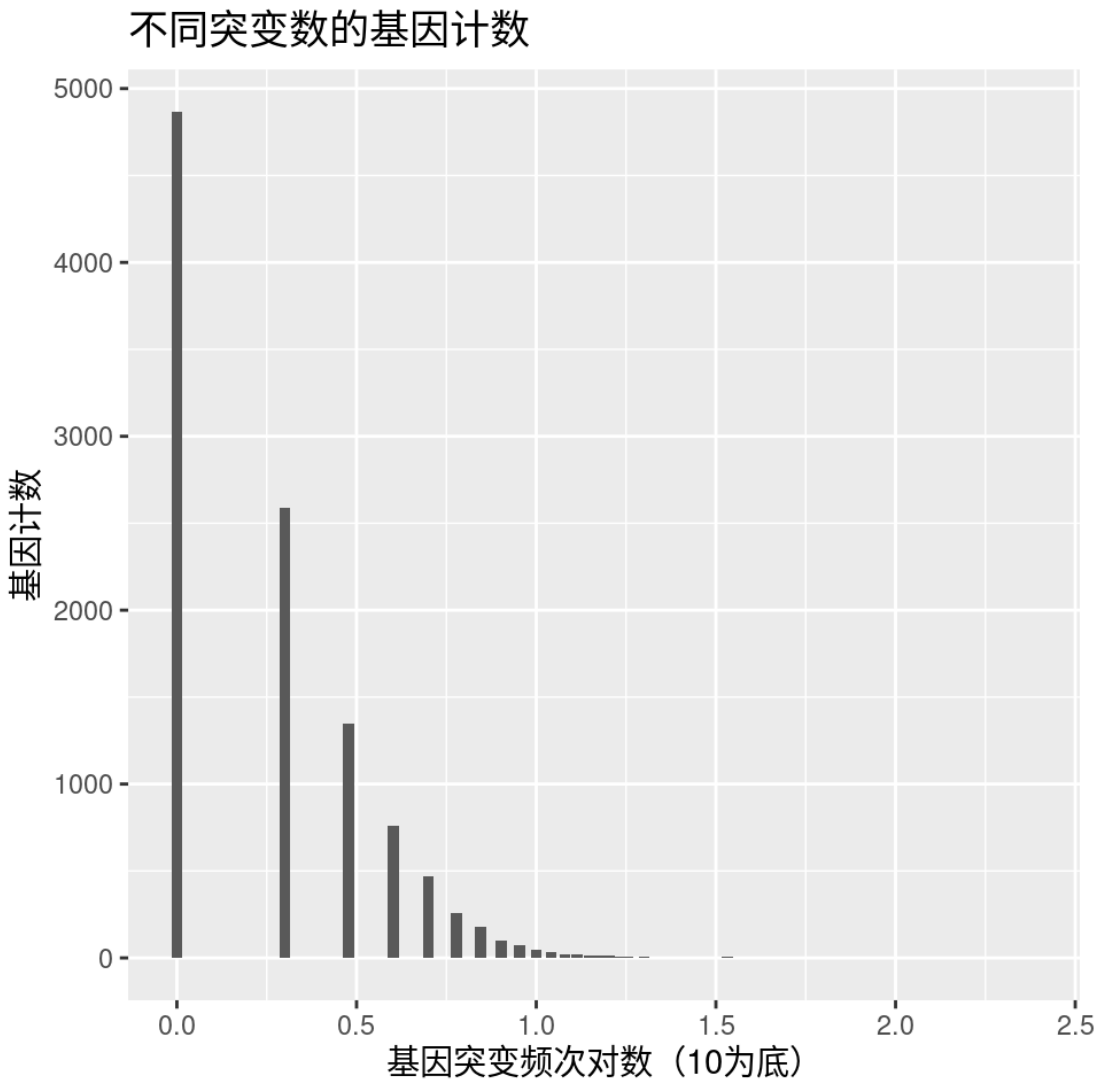
基因突变频次分析

我们可以对使用 refGene 产生的注释文件进行针对基因的计数，并分析各个基因发生突变的频次。

```
cat tbl_anno.refGene.exonic_variant_function | \
cut -f 3 | \
cut -d ":" -f 1 | \
sort | \
uniq -c | \
sort -rn > \
mutation_count_per_gene.txt
```

可以看到除去未知基因名的位置上发生的突变，突变最多的是 MUC4 基因，其次是 OR8U1。

```r
library(ggplot2)
genesVariationCount$logFreq <- log10(genesVariationCount$Freq)
genesVariationCount %>%
    ggplot(aes(x = logFreq)) +
    geom_bar(width = 0.03) +
    labs(
        x = "基因突变频次对数（10为底）",
        y = "基因计数",
        title = "不同突变数的基因计数"
    ) +
    theme(
        legend.position = "right"
    )
```



不难发现，在对数作图中，我们所观察到的基因计数随着突变数的上升迅速下降，显然不符合泊松分布在大数定律下的近似，因此可以推测大部分基因上的突变是受限的，发生多个突变的概率显著下降。可能原因为多个突变会导致基因功能的丧失，进而在选择压力下无法保留这样的突变。

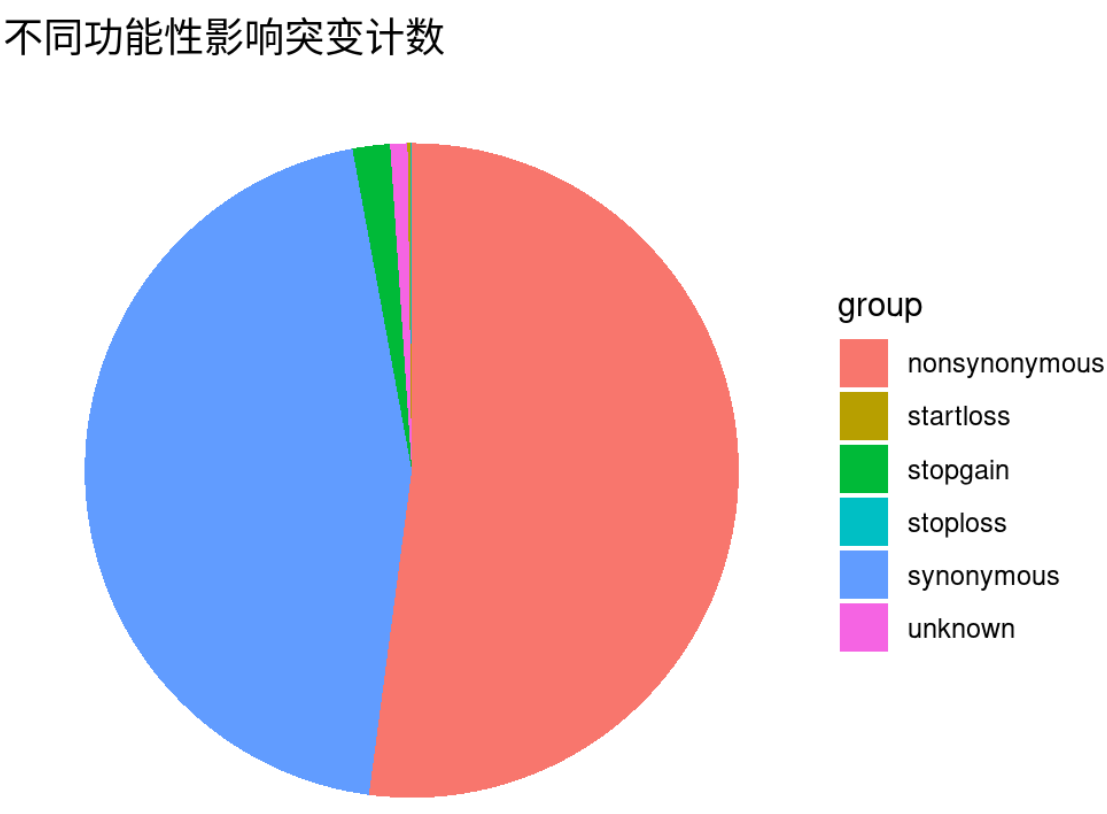
基因外显子区域突变的影响

我们可以统计注释文件中功能性影响的字段来进行简单的统计。

```
cat tbl_anno.refGene.exonic_variant_function | \
cut -f 2 | \
sort | \
uniq -c | \
sort -rn > \
functional_effects.txt
```

并且作饼图来进行可视化

```
library(ggforce)
cumFreq <- cumsum(funcConseq$Freq)
data.frame(
    ymin = cumFreq - funcConseq$Freq + .01,
    ymax = cumFreq - .01,
    group = funcConseq$Consequences
) %>%
    ggplot() +
    geom_rect(aes(
        xmin = 0,
        xmax = 1,
        ymin = ymin,
        ymax = ymax,
        fill = group
    )) +
    coord_polar(theta = "y") +
    theme_void() +
    labs(title = "不同功能性影响突变计数")
```



从图中不难发现，绝大多数突变为同义/非同义突变，而对于起始缺失、终止缺失、提前终止的突变类型都非常少。从实际情况考虑，后三种类型突变对于蛋白结构的影响一般大于前两者，因此在选择压力下更难保留，因而频次显著更低。

结论

通过批量 SNP 注释，我们可以推测 SNP 本身的功能，可以从与例如 GWAS 等研究方法相反的方向，从基因出发推测功能与表型，进行分析交叉验证，进而为这些研究提供更多思路与见解。

参考资料

[1] WIKIPEDIA. Single-nucleotide polymorphism[Z/OL](2021–05). https://en.wikipedia.org/wiki/Single-nucleotide_polymorphism.

[2] TEAMA S. DNA Polymorphisms: DNA-Based Molecular Markers and Their Application in Medicine[M/OL]. https://www.intechopen.com/books/genetic-diversity-and-disease-susceptibility/dna-polymorphisms-dna-based-molecular-markers-and-their-application-in-medicine. DOI:10.5772/intechopen.79517.

[3] WANG K, LI M, HAKONARSON H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data[J/OL]. Nucleic Acids Research, 2010, 38(16): e164–e164. https://doi.org/10.1093/nar/gkq603. DOI:10.1093/nar/gkq603.

[4] DANECEK P, BONFIELD J K, LIDDLE J, 等. Twelve years of SAMtools and BCFtools[J]. GigaScience, 2021, 10(2). DOI:10.1093/gigascience/giab008.