

# Maize Genome Database

Lingbin Ni

## 【项目地址】

</home/Lingbin/maizegenome/maizedbfarm>

## 【项目说明】

### 1. 操作系统

Ubuntu 16.04 LTS


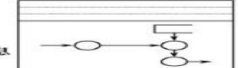

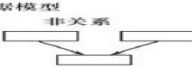
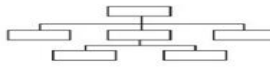


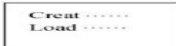

### 2. 数据库管理系统安装

```
wget --output-document=- https://www.monetdb.org/downloads/MonetDB-GPG-KEY |
sudo apt-key add -
sudo apt update
sudo apt install monetdb5-sql monetdb-client
```

### 3. 数据库创建

```
shell> monetdbd create ~/dbfarmname
shell> monetdbd start ~/dbfarmname
shell> monetdb create dbname
shell> monetdb release dbname
shell> monetdb start dbname
shell> mclient -u monetdb -d dbname
```

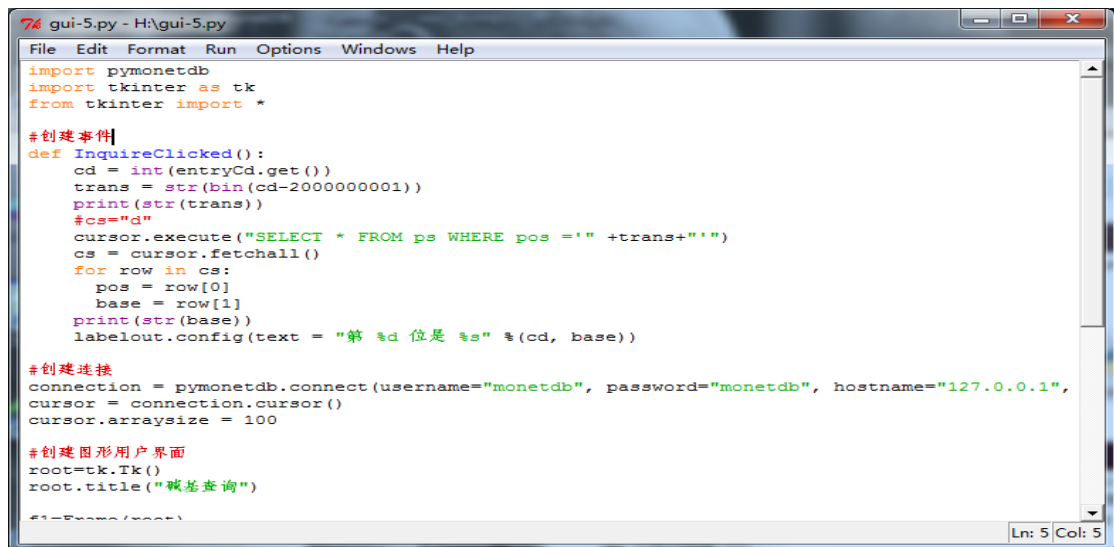
### 4. 数据库结构设计

设计阶段	设计描述	
	数 据	处 理
需求分析	数据字典、全系统中数据项、数据流、数据存储的描述	数据流图和判定表（判定树）、数据字典中处理过程的描述
概念结构设计	概念模型（E-R图）  数据字典	系统说明书包括： ① 新系统要求、方案和概图 ② 反映新系统信息流的数据流图 
逻辑结构设计	某种数据模型 关系  非关系 	系统结构图（模块结构） 
物理设计	存储安排 方法选择 存取路径建立 	模块设计 IPO 表 
数据库实施阶段	编写模式 装入数据 数据库试运行 	程序编码、编译联结、测试 
数据库运行和维护	性能监测、转储/恢复 数据库重组和重构	新旧系统转换、运行、维护（修正性、适应性、改善性维护）

### 5. 数据导入

- a) 获取数据
- b) 确定导入格式
- c) 格式化
- d) 批量导入

## 6. 数据管理查询



```
gui-5.py - H:\gui-5.py
File Edit Format Run Options Windows Help

import pymonetdb
import tkinter as tk
from tkinter import *

#创建事件
def InquireClicked():
    cd = int(entryCd.get())
    trans = str(bin(cd-2000000001))
    print(str(trans))
    #cs="d"
    cursor.execute("SELECT * FROM ps WHERE pos ='"+trans+"'")
    cs = cursor.fetchall()
    for row in cs:
        pos = row[0]
        base = row[1]
        print(str(base))
        labelout.config(text = "第 %d 位是 %s" %(cd, base))

#创建连接
connection = pymonetdb.connect(username="monetdb", password="monetdb", hostname="127.0.0.1",
cursor = connection.cursor()
cursor.arraysize = 100

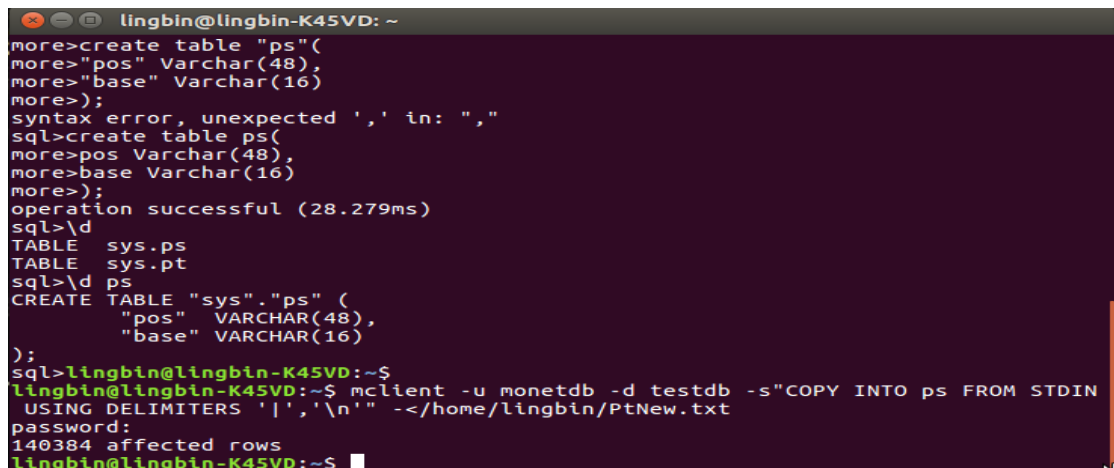
#创建图形用户界面
root=tk.Tk()
root.title("碱基查询")

#主循环
root.mainloop()
```

## 【项目展示】

### 1. 数据库 Table 的创建及键的定义

在目标数据库中创建表格“ps”，该表格包含位置“pos”和碱基“base”两列，定义数据格式，并将主键定义为“pos”。



```
lingbin@lingbin-K45VD: ~
more>create table "ps"(
more>"pos" Varchar(48),
more>"base" Varchar(16)
more>);
syntax error, unexpected ',' in: ",
sql>create table ps(
more>pos Varchar(48),
more>base Varchar(16)
more>);
operation successful (28.279ms)
sql>\d
TABLE sys.ps
TABLE sys.pt
sql>\d ps
CREATE TABLE "sys"."ps" (
    "pos" VARCHAR(48),
    "base" VARCHAR(16)
);
sql>lingbin@lingbin-K45VD:~$
lingbin@lingbin-K45VD:~$ mclient -u monetdb -d testdb -s"COPY INTO ps FROM STDIN
USING DELIMITERS '|' '\n'" -</home/lingbin/PtNew.txt
password:
140384 affected rows
lingbin@lingbin-K45VD:~$
```

## 2. 数据导入及管理

导入的测试数据为玉米 pt 染色体，该染色体包含 140384 个碱基。值得注意的是，为了方便后续大量数据导入及编码，我们将碱基的编码从 -2000000000 开始并将其转换成二进制数字。具体程序可见 Format\_File.py。

```
lingbin@lingbin-K45VD: ~
sql>\d ps
CREATE TABLE "sys"."ps" (
  "pos" VARCHAR(48),
  "base" VARCHAR(16)
);
sql>select count(*) from ps;
+-----+
| L3      |
+=====+
| 140384  |
+-----+
1 tuple (2.730ms)
sql>select base,count(*) as total
more>from ps GROUP BY base ORDER BY base;
+-----+-----+
| base | total |
+=====+=====+
| A     | 43281 |
| C     | 26908 |
| G     | 27087 |
| T     | 43108 |
+-----+-----+
4 tuples (8.343ms)
sql>
```

## 3. 通过 API 实现 Python 操纵

通过建立 API，实现了通过 Python 而非 SQL 语言操作数据库。下图为简单的数据库操作。具体程序可见 API\_test.py。

```
lingbin@lingbin-K45VD: ~
(pytest) lingbin@lingbin-K45VD:~$ pip install pymonetdb
Collecting pymonetdb
  Using cached pymonetdb-1.0.6-py2.py3-none-any.whl
Collecting six (from pymonetdb)
  Using cached six-1.10.0-py2.py3-none-any.whl
Installing collected packages: six, pymonetdb
Successfully installed pymonetdb-1.0.6 six-1.10.0
(pytest) lingbin@lingbin-K45VD:~$ python2
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pymonetdb
>>> connection=pymonetdb.connect(username="monetdb",password="monetdb",hostname=
"127.0.0.1",database="testdb")
>>> cursor=connection.cursor()
>>> cursor.arraysize=100
>>> cursor.execute('SELECT * FROM ps')
140384
>>> cursor.fetchone()
(u'-0b1110111001101011001010000000000', u'G\r')
>>> cursor.description
[('pos', 'varchar', None, 48, None, None, None), ('base', 'varchar', None, 16, N
one, None, None)]
>>>
```

#### 4. 通过自定义 GUI 实现数据库查询

通过编写自定义 GUI 程序, 实现了通过图形用户界面对数据库的碱基位置进行查询。具体程序可见 Monet\_gui.py。

```
lingbin@lingbin-K45VD: ~  
lingbin@lingbin-K45VD:~$ source /home/lingbin/maizegenome/maizedbfarm/py3/bin/activate  
(py3) lingbin@lingbin-K45VD:~$ python /home/lingbin/gui-5.py-0b11101110011010110  
01001111111100  
T  
-0b1110111001101011001001111010100  
G  
-0b1110111001101010110001010010011  
T  
(py3) lingbin@lingbin-K45VD:~$ python /home/lingbin/gui-5.py  
-0b1110111001101010010110001010111  
C  
□
```

