基因排序器的设计

代码之美第13章

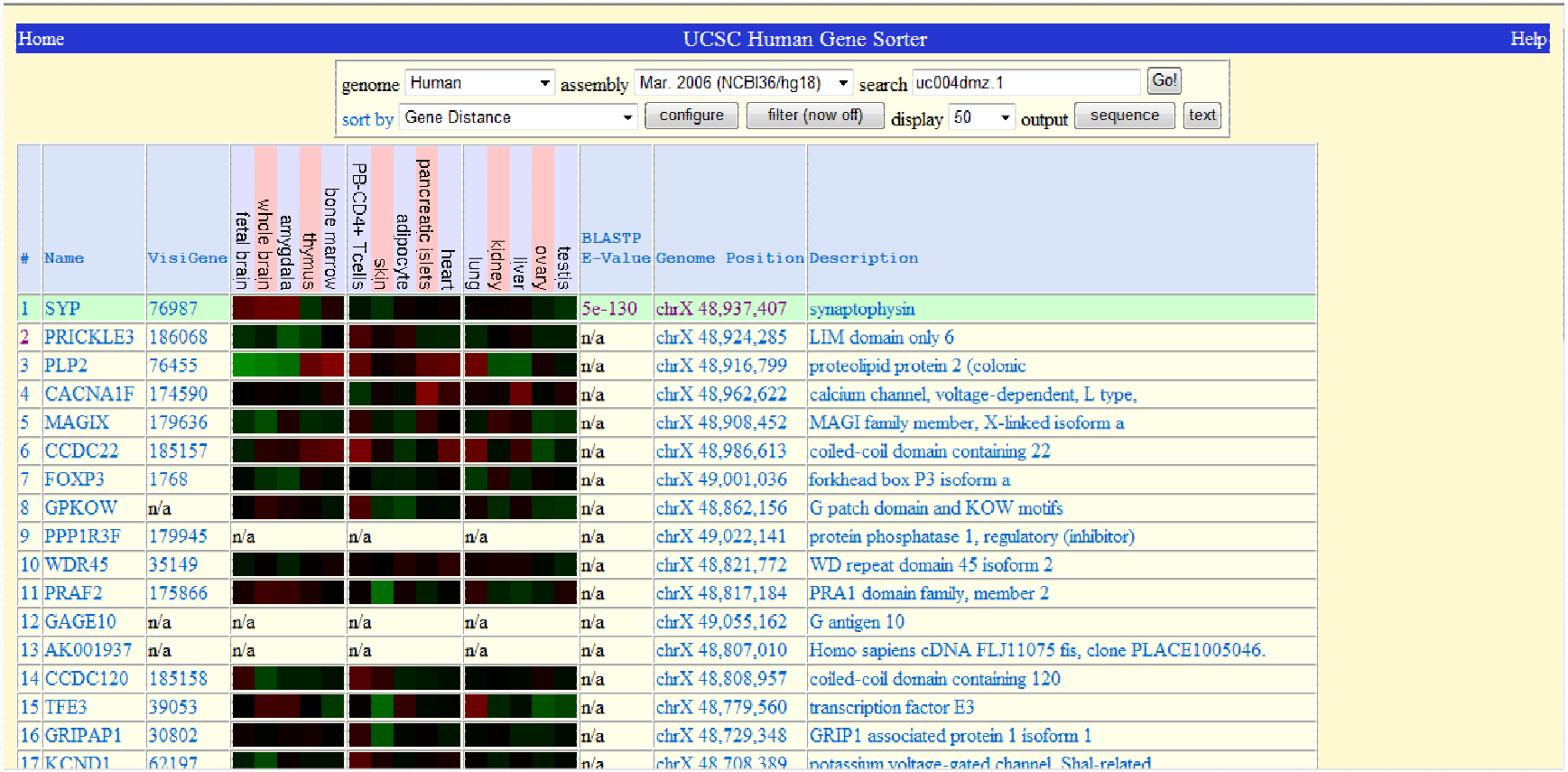
基因排序器

- 网址 http://genome.ucsc.edu/cgi-bin/hgNear
- 用于处理人类基因组项目(Human Genome Project)中产生的数据

处理流程

- 1 通过CGI收集收集来自用户的输入
- 2 查询MySQL数据库
- 3 用HTML呈现结果

用户界面



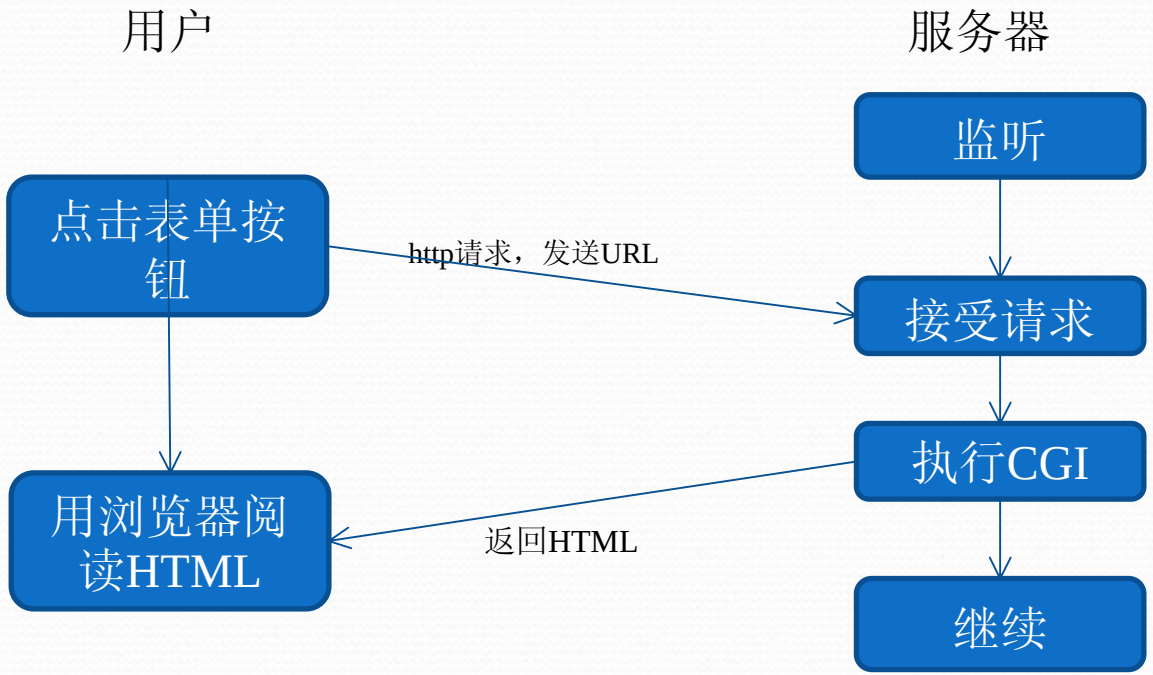
Home | UCSC Human Gene Sorter | Help

genome Human assembly Mar. 2006 (NCBI36/hg18) search uc004dmz.1 Go!

sort by Gene Distance configure filter (now off) display 50 output sequence text

#	Name	VisiGene	fetal brain	whole brain	amygdala	thymus	bone marrow	PB-CD4+ Tcells	skin	adipocyte	pancreatic islets	heart	lung	kidney	liver	ovary	testis	BLASTP E-Value	Genome Position	Description
1	SYP	76987																5e-130	chrX 48,937,407	synaptophysin
2	PRICKLE3	186068																n/a	chrX 48,924,285	LIM domain only 6
3	PLP2	76455																n/a	chrX 48,916,799	proteolipid protein 2 (colonic
4	CACNA1F	174590																n/a	chrX 48,962,622	calcium channel, voltage-dependent, L type,
5	MAGIX	179636																n/a	chrX 48,908,452	MAGI family member, X-linked isoform a
6	CCDC22	185157																n/a	chrX 48,986,613	coiled-coil domain containing 22
7	FOXP3	1768																n/a	chrX 49,001,036	forkhead box P3 isoform a
8	GPKOW	n/a																n/a	chrX 48,862,156	G patch domain and KOW motifs
9	PPP1R3F	179945	n/a					n/a					n/a					n/a	chrX 49,022,141	protein phosphatase 1, regulatory (inhibitor)
10	WDR45	35149																n/a	chrX 48,821,772	WD repeat domain 45 isoform 2
11	PRAF2	175866																n/a	chrX 48,817,184	PRA1 domain family, member 2
12	GAGE10	n/a	n/a					n/a					n/a					n/a	chrX 49,055,162	G antigen 10
13	AK001937	n/a	n/a					n/a					n/a					n/a	chrX 48,807,010	Homo sapiens cDNA FLJ11075 fis, clone PLACE1005046.
14	CCDC120	185158																n/a	chrX 48,808,957	coiled-coil domain containing 120
15	TFE3	39053																n/a	chrX 48,779,560	transcription factor E3
16	GRIPAP1	30802																n/a	chrX 48,729,348	GRIP1 associated protein 1 isoform 1
17	KCND1	62197																n/a	chrX 48,708,389	potassium voltage-gated channel, Shal-related

通过Web与用户保持对话



CGI

- 优点：
- 可移植性高

- 缺点：
- 交互性一般

生命周期短暂的CGI脚本

- 复杂的CGI脚本需要能长期保存数据
- 方法一：隐藏的CGI变量（数据通过隐藏的<INPUT>标签保存在HTML中），生命周期为单次会话。
- 方法二：cookie，生命周期很长。

行李车

- 维护两张数据库表，一张关联用户，另一张关联会话。
- 表的格式：

  一个主键列，一个blob字段，一个跟踪使用时间和访问计数的字段。（Blob：包含通过URL传送的所有变量/值对。）

- Cookie指向用户表
- 隐藏的CGI变量指向会话表

行李车

- 网站上所有的CGI脚本共享同一个一个行李车。

- 避免变量名冲突
  CGI变量名使用CGI脚本的名字开头。

多态

- 在C语言中构造多态对象：
- 1 用结构体代替对象
- 2 用函数指针代替多态方法

column结构体

```
struct column
/*大表格中的一列，hgNear中的核心数据结构*/
{
        /*所有的列都存在的数据；*/
        struct  column *next;   /*链表中的下一列*/
        char  *name;             /*列名，用户看不到*/
        char  *shortLabel;       /*列标签*/
        char  *shortLabel;       /*列描述*/

        /*方法*/
        void  (*cellPrint)(struct column *col, struct genePos *gp, struct sqlConnection *conn);
        /*在HTML中打印该列的一个单元格*/

        void (*labelPrint)(struct column *col);
        /*在标签行中打印标签*/
        void (*filterControls)(struct column *col,  struct sqlConnection *conn);
        /*打印高级过滤器中的控件标签*/
        struct genePos *(*advFilter)(struct column *col, struct sqlConnection *conn)
        /*返回高级过滤器的位置列表*/

        /*下面的几个字段是查找表(Lookup table)使用的*/
        char *table;/*关联表(associated table)的名字*/
        char *keyField;/*关键表中的GeneID字段*/
        char *valField;/*关键表中的Value字段*/

        /*除了跟查找相关的字段，关联表还使用如下字段*/
        char *queryFull; /*返回两列键/值的查询*/
        char *queryOne;/*给定键，返回值相关值的查询*/
        char *invQueryOne;/*给定值，返回值相关键的查询*/
};
```

创建列对象的方法

- 基于columnDb.ra文件创建
- columnDb记录包含一些字段用于描述列名，用户可见的短标签和长标签，列在表格中的默认位置，默认是否可见，以及一些类型字段。
- 类型字段决定列拥有哪些方法。
- 除此之外，可能存在类型特定的附加字段。
- 在很多情形中，columnDb包含了查询数据库表的sql语句以及链接到列中每个数据项的URL。

columnDb.ra

name proteinName
shortLable UniProt
longLable UniProt (SwissProt/TrEMBL) Protein Display ID
priority 2.1
visibility off
type association kgXref
queryFull select kgID, spDisplayID from kgXref
queryOne select spDisplayID, spID from kgXref where kgID = ‘%s’
invQueryOne select kgID from kgXref where spDisplayID = ‘%s’
search fuzzy
itemUrl http://us.expasy.org/cgi-bin/niceprot.pl?%s

name proteinAcc
shortLable UniProt Acc
longLable UniProt (SwissProt/TrEMBL) Protein Accession
priority 2.15
visibility off
type lookup kgXref kgID spID
search exact
itemUrl http://us.expasy.org/cgi-bin/niceprot.pl?%s

name refSeq
shortLable RefSeq
longLable NCBI RefSeq Gene Accession
prirority 2.2
visibility off
Type lookup knowToRefSeq name value
search exact
itemUrl http://www.ncbi.nlm.nih.gov/entrez/query.
fcgi?cmd=Search&db=Nucleotide&term=%s&doptcmdl=GenBank&tool=genome.ucsc.edu

- columnDb被编排成三层目录结构。
- 根目录：所有物种都会出现的列信息。
- 中间层：特定物种的信息。
- 底层：特定集合的信息（将特定物中的DNA分成若干集合）

列的类型

- 1 查找类型
- 引用一张表，对GeneID进行索引
- 包含表、geneID字段和列中显示的字段

- 2 关联类型
- 一个基因关联多个值

- 3 其他
- 如基因表达的列
- 相对复杂

滤除无关的基因

- 每一列有两个过滤方法：用来向过滤器用户界面输出HTML的filterControls和完成实际的过滤的advFilter。
- 这两个方法通过行李车变量进行交互。
- 过滤器被排成一个链。把每个过滤器依次调用一遍。

过滤器的性能

- 在每个基因上用的时间不超过万分之一秒。
- 避免磁盘寻道。
  先到行李车检查是否设置过相关变量。
  每次读取整张表。

大规模美丽代码理论

- 提高代码的可理解性
- 合适的名字

```
static struct genePos *wildAssociationFilter(
        struct slName *wildList, boolean orLogic, struct column *col,
        struct sqlConnection *conn, struct genePos *list)
/*过滤匹配通配符列表中的任何一项的关联.*/
{
/*将关联按gene ID分组*/
struct assocGroup *ag = assocGroupNew(16);
struct sqlResult *sr = sqlGetResult(conn, col->queryFull);
char **row;
while ((row = sqlNextRow(sr)) != NULL)
      assocGroupAdd(ag, row[0], row[1]);
sqlFreeResult(&sr);

/*寻找匹配的关联并把它们放进passHash表中.*/
struct hash *passHash = newHash(16); /*哈希表，用于保存通过过滤的项目*/
struct genePos *gp;
for (gp = list; gp != NULL; gp = gp->next)
     {
      char *key = (col->protKey ? gp->protein : gp->name);
      struct assocList *al = hashFindVal(ag->listHash, key);
      if ( al != NULL)
            {
            if (wildMatchRefs(wildList, al-> list, orLogic))
                  hashAdd(passHash, gp->name, gp);
             }
       }
/*创建经过过滤的列表，善后清理，返回. */
list = weedUnlessInhash(list, passHash);
hashFree(&passHash);
assocGroupFree(&ag);
return list;
}
```

其他提高代码可理解性的方法

- 尽可能让作用域局部化。
- 减小副作用