TelCentSpec 技术交底书

专利申请交底书

专利权人：曹昕
联系人：曹昕
联系电话：+86 13812345678
快递地址：北京市海淀区 XX 路 XX 号
电子邮箱：caoxin@example.com
填报日期：2024 年 10 月 12 日
发明人名单：曹昕
第一发明人身份证号：110101198001012345

专利名称

TelCentSpec: Chromosome-Specific Telomere and Centromere Repeats Detection and Specificity Filtering System

一、技术领域

本发明属于生物信息学和基因组学技术领域，具体涉及一种用于检测染色体特异性端粒和着丝粒重复序列的探针筛选系统。该探针筛选系统旨在通过基因拷贝数的变化，进行遗传疾病的诊断。通常，人体内每个基因位点有 2 个拷贝，分别来自父母。本发明的探针设计有助于检测染色体上基因拷贝数的变化，这对诊断涉及基因拷贝数异常的遗传疾病具有重要意义。

二、现有技术分析

目前在基因组学和分子生物学领域中，已有一些用于探针寻找的工具程序，如 BLAST 等。然而，这些工具在筛选特异性探针时存在明显不足，主要表现在以下几个方面：

1. ** 探针特异性不足 **：现有的探针筛选工具无法有效区分目标染色体和非目标染色体上的相似序列，导致探针在非目标染色体上也能发生结合，从而影响检测结果的准确性。

2. ** 非 fold 空间结构未被考虑 **：在探针筛选中，现有技术通常忽略了探针的空间结构稳定性。探针在结合靶标序列时，如果自身形成 fold 结构，会显著降低其与靶标的结合效率，进而影响探针的灵敏度和特异性。

3. ** 筛选效率低下 **：目前的工具程序多次筛选探针的效率较低，无法在短时间内生成高特异性的探针。

因此，亟需一种能够同时提高探针特异性、优化探针结构，并提高筛选效率的算法和工具，以应对当前基因组检测中对高特异性探针的迫切需求。

三、需解决的关键技术说明

本发明的关键技术点在于：1. ** 特异性问题的解决 **：通过两轮比对策略，其中第二轮比对将探针与非目标染色体上的序列进行比对，从而有效剔除非目标染色体上具有相似性的探针，显著提高探针对目标染色体的特异性。2. ** 比对阈值的优化 **：经过多次实验，本发明调整了两轮比对的阈值，使得在保证特异性的同时，筛选出的探针具备更高的灵敏度。第一轮比对的阈值设定为 0.8，第二轮比对的阈值设定为 0.3，通过这些阈值的优化实现了探针效果的最优化。3. ** 非 fold 结构的应用 **：为进一步提高探针的性能，本发明引入了非 fold 结构的预测，确保筛选出的探针在靶标序列上具有稳定的结合性，而不会因为探针自身的折叠结构影响检测结果。

四、发明内容

1. **技术方案**：

本发明提出了一种染色体特异性的端粒和着丝粒重复序列检测及筛选系统。系统分为探针生成、比对筛选、结构预测和重复筛选四个主要步骤。

以下为具体的算法公式部分：

1　1. 探针生成

在探针生成模块中，TelCentSpec 从目标染色体的着丝粒和端粒区域截取序列，生成探针。假设染色体序列为 S，目标染色体的着丝粒和端粒区域的起始和结束位置分别为 $[start, end]$，则生成的探针可以表示为：

$$P = S[start : end] \tag{1}$$

根据提供的文件，染色体的起始和结束坐标可以设定为（示例）：

$$P = S[43044294 : 43125364] \quad \text{(例如：BRCA1 基因在染色体 17 上的坐标)}$$

其中，P 是从染色体序列 S 中截取的片段，作为探针。

对于多个探针生成的情况，有：

$$P_i = S_i[start : end], \quad (i = 1, 2, \ldots, n) \tag{2}$$

其中 n 是生成的探针数。

2 2. 第一轮比对

探针生成后，使用 BLAST 工具将探针与目标染色体的着丝粒和端粒区域进行比对。探针与目标染色体的相似度定义为 $S_{target}(P)$。第一轮比对的目的是筛选出与目标区域相似度高于某个阈值 θ_1 的探针：

$$S_{target}(P_i) \geq \theta_1 \tag{3}$$

假设相似度阈值 θ_1 为 0.8，即要求探针与目标染色体的比对相似度不低于 80

3 3. 第二轮比对

在第二轮比对中，筛选出的探针与非目标染色体进行比对，剔除那些在非目标染色体上也具有高相似度的探针。非目标染色体的相似度表示为 $S_{non-target}(P)$。筛选条件是相似度低于阈值 θ_2：

$$S_{non-target}(P_i) \leq \theta_2 \tag{4}$$

假设非目标染色体上的相似度阈值 θ_2 为 0.3，即要求探针在非目标染色体上的相似度低于 30

4 4. 结构预测与筛选

对于通过两轮比对的探针，我们还需要通过非 fold 结构预测工具评估探针的稳定性。假设探针的非 fold 稳定性得分为 $F(P)$，则筛选条件为：

$$F(P_i) \geq \theta_3 \tag{5}$$

假设非 fold 结构的稳定性得分阈值 θ_3 为 0.7。

此外，我们还根据探针在目标和非目标区域的重复次数进行筛选。定义探针在目标区域的重复次数为 $R_{target}(P)$，在非目标区域的重复次数为 $R_{non-target}(P)$，筛选条件为：

$$R_{target}(P_i) \geq r_1 \quad \text{且} \quad R_{non-target}(P_i) \leq r_2 \tag{6}$$

其中，r_1 可以设定为 5 次，r_2 可以设定为 1 次，即探针在目标区域的重复次数至少为 5，而在非目标区域的重复次数不超过 1 次。

5 5. 最终筛选公式

最终，筛选出的探针需要同时满足以下所有条件：

1. 第一轮比对：$S_{target}(P_i) \geq \theta_1$（设定 $\theta_1 = 0.8$）
2. 第二轮比对：$S_{non-target}(P_i) \leq \theta_2$（设定 $\theta_2 = 0.3$）
3. 非 fold 结构预测：$F(P_i) \geq \theta_3$（设定 $\theta_3 = 0.7$）
4. 重复次数筛选：$R_{target}(P_i) \geq r_1$ 且 $\mathrm{R}_{non-target}(P_i) \leq r_2$（设定 $r_1 = 5, r_2 = 1$）

所有满足这些条件的探针 P_i 将被选为最终的合格探针。

2. **实施效果**：

通过该系统，可以显著提高染色体特异性探针的筛选效率和特异性，尤其是在端粒和着丝粒区域的探针筛选中。本发明提出的算法有效降低了非目标染色体上相似序列的干扰，能够提供高质量的探针设计，为基因组研究和临床应用提供了重要的工具支持。

五、发明实施方式

本发明的实施依赖于 Linux 操作系统和 Python 编程语言。在 Linux 系统中，本发明使用 BLAST 工具进行序列比对，同时通过 Python 脚本进行探针生成和筛选。

1．环境要求：

- 操作系统：Linux (Ubuntu 18.04 或更新版本)
- 编程语言：Python 3.6 及以上版本
- 所需工具：BLAST+ 2.9.0 或更高版本
- 其他依赖：Biopython, Numpy, Scipy
- 硬件资源：建议使用具有至少 16 GB 内存和 4 核处理器的服务器环境进行大规模探针筛选

2．输入文件：本系统的输入文件主要为目标染色体的 FASTA 格式基因组序列，以及非目标染色体的基因组序列库，用于进行两轮比对。文件格式要求如下：

- 目标染色体序列文件（FASTA 格式）
- 非目标染色体序列库（FASTA 格式）

3．执行步骤：

1. 在 Linux 系统中，首先通过以下命令安装所需的依赖包：

```
sudo apt-get update
sudo apt-get install ncbi-blast+ python3-pip
pip3 install biopython numpy scipy
```

2. 使用 Python 脚本生成初步探针序列：

```
python3 IndexCreator.py --input target_chromosome.fasta --output probes.fasta
```

3. 使用 BLAST 工具进行第一轮比对，将生成的探针序列与目标染色体进行比对：

```
blastn -query probes.fasta -db target_chromosome_db -out results_target.txt
```

4. 对筛选出的探针进行第二轮比对，将探针与非目标染色体库进行比对：

```
blastn -query probes.fasta -db nontarget_chromosome_db -out results_nontarget.txt
```

5. 根据比对结果，使用 Python 脚本进行非 fold 结构筛选并优化探针性能：

```
python3 GENOME_FILTER.py --input results_target.txt --filter results_nontarget.txt
```

6. 经过两轮比对和筛选后，输出最终筛选的特异性探针序列：

```
python3 FinalProbeSelector.py --input filtered_probes.fasta --output final_probes.fasta
```

4．所需计算资源：根据本系统的大规模探针筛选需求，建议使用具有以下配置的服务器环境：

- 处理器：4 核或以上
- 内存：至少 16 GB
- 硬盘：至少 500 GB 可用存储空间，用于存储比对数据库和中间文件