“乐孕安”优生优育产品

染色体解析：助力新一代试管婴儿技术

张洋

乐土精准医疗

2017 年 7 月 17 日

目　　录

产品概览

本产品包括两个阶段

1. 查因
   通过高精度的全基因组测序，大数据的分析，检测样本的染色体异常情况，并给出异常断点的验证策略，零假阳性。
   最终给出临床表征的遗传学解释。
2. 助孕
   新一代试管婴儿技术，充分利用第一阶段的检测结果“PCR 引物对”，完美阻断亲代结构异常向子代遗传。再配合 PGS 剔除自发突变产生的不良胚胎。
   最终提高植入成功率，降低出生缺陷，给家庭一个健康的宝宝。

“乐孕安” 第一阶段

查因

定义

A prime number is a number that has exactly two divisors.

例

- 2 is prime (two divisors: 1 and 2).

“乐孕安” 第一阶段

查因

定义

A prime number is a number that has exactly two divisors.

例

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).

“乐孕安” 第一阶段

查因

定义

A prime number is a number that has exactly two divisors.

例

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).
- 4 is not prime (three divisors: 1, 2, and 4).

There Is No Largest Prime Number

The proof uses *reductio ad absurdum*.

定理

There is no largest prime number.

证明.

1. Suppose p were the largest prime number.

4. But $q + 1$ is greater than 1, thus divisible by some prime number not in the first p numbers. □

There Is No Largest Prime Number

The proof uses *reductio ad absurdum*.

定理

There is no largest prime number.

证明.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.

4. But $q + 1$ is greater than 1, thus divisible by some prime number not in the first p numbers. □

There Is No Largest Prime Number

The proof uses *reductio ad absurdum*.

定理

There is no largest prime number.

证明.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. But $q + 1$ is greater than 1, thus divisible by some prime number not in the first p numbers. □

There Is No Largest Prime Number

The proof uses *reductio ad absurdum*.

定理

There is no largest prime number.

证明.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. But $q + 1$ is greater than 1, thus divisible by some prime number not in the first p numbers. □

The proof used *reductio ad absurdum*.

What' s Still To Do?

Answered Questions

How many primes are there?

Open Questions

Is every even number the sum of two primes?

What's Still To Do?

- Answered Questions
  - How many primes are there?
- Open Questions
  - Is every even number the sum of two primes?

What's Still To Do?

Answered Questions

How many primes are there?

Open Questions

Is every even number the sum of two primes?[1]


[Goldbach, 1742] Christian Goldbach.
A problem we should try to solve before the ISPN ′ 43 deadline, Letter to Leonhard Euler, 1742.

An Algorithm For Finding Prime Numbers.

```
int main (void)
{
std::vector<bool> is_prime (100, true);
for (int i = 2; i < 100; i++)
if (is_prime[i])
{
std::cout << i << " ";
for (int j = i; j < 100; is_prime [j] = false, j+=i);
}
return 0;
}
```

An Algorithm For Finding Prime Numbers.

```
int main (void)
{
std::vector<bool> is_prime (100, true);
for (int i = 2; i < 100; i++)
if (is_prime[i])
{
std::cout << i << " ";
for (int j = i; j < 100; is_prime [j] = false, j+=i);
}
return 0;
}
```

Note the use of `std::`.

An Algorithm For Finding Primes Numbers.

```
^^Iint main (void)
^^I{
^^I std::vector<bool> is_prime (100, true);
^^I for (int i = 2; i < 100; i++)
^^I
^^I
^^I
^^I
^^I
^^I
^^I return 0;
^^I}
```

An Algorithm For Finding Primes Numbers.

```
^^Iint main (void)
^^I{
^^I std::vector<bool> is_prime (100, true);
^^I for (int i = 2; i < 100; i++)
^^I if (is_prime[i])
^^I {
^^I
^^I
^^I
^^I }
^^I return 0;
^^I}
```

An Algorithm For Finding Primes Numbers.

```
^^Iint main (void)
^^I{
^^I std::vector<bool> is_prime (100, true);
^^I for (int i = 2; i < 100; i++)
^^I if (is_prime[i])
^^I {
^^I std::cout << i << " ";
^^I for (int j = i; j < 100;
^^I is_prime [j] = false, j+=i);
^^I }
^^I return 0;
^^I}
```

An Algorithm For Finding Primes Numbers.

```
^^Iint main (void)
^^I{
^^I std::vector<bool> is_prime (100, true);
^^I for (int i = 2; i < 100; i++)
^^I if (is_prime[i])
^^I {
^^I std::cout << i << " ";
^^I for (int j = i; j < 100;
^^I is_prime [j] = false, j+=i);
^^I }
^^I return 0;
^^I}
```

Note the use of std::.