

Lacuna Dev Ingresso

1. Introdução

Lacuna Software acaba de iniciar um ramo fictício: Lacuna Genetics

Para melhorar nossa pesquisa e poder de computação, decidimos distribuir algumas operações de computação.

Seu trabalho é seguir a documentação abaixo e criar um programa C# .NET que se comunique com nossas APIs e nos ajude a concluir as operações de DNA.

Depois de se inscrever no teste, você tem 7 dias para concluí-lo.

Depois de concluído, envie-nos seu projeto final em arquivo ZIP e seu Currículo (Currículo) para `admissions@lacunasoftware.com`.

Se você tiver alguma dúvida durante o teste, entre em contato conosco pelo mesmo endereço de e-mail.

Nota: Isto não é uma corrida! Certifique-se de mostrar suas habilidades de codificação, conhecimento sobre modularização, serialização de dados e reutilização de código. Divirta-se!

2. Inscreva-se

Você precisará lidar com informações ultrassecretas, então o primeiro passo é criar um usuário no sistema e solicitar um AccessToken para comunicação com as APIs autenticadas.

Base address: <https://gene.lacuna.cc/>

2.1. Criar usuário

'[POST] /api/users/create'

Request body

```
{
  // Allowed a-z, A-Z and 0-9 chars only
  // Min size 4 chars and max size 32 chars
  username: string,

  // Your email address so we are able to contact you
  email: string,

  // Min size 8 chars
  password: string //
}
```

Response

```
{
  code: string, // ['Success', 'Error']
  message?: string
}
```

2.2 Solicitar token de acesso

'[POST] /api/users/login'

Request body

```
{
  username: string,
  password: string
}
```

Response

```
{
  accessToken?: string,
  code: string, // ['Success', 'Error']
  message?: string
}
```

Se tudo estiver OK, você receberá um `Success` código de resposta e uma `AccessToken` string. O token de acesso deve ser usado no `Authorization` parâmetro Header como um esquema de token de portador OAuth.

O token de acesso é válido por 2 minutos, se expirado as APIs autenticadas retornarão um `Unauthorized` código de resposta com a mensagem: “Bad token: token is expired”, será necessário solicitar um novo.

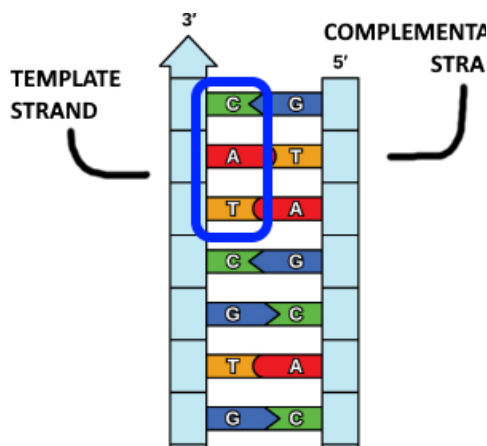
3. Revisão rápida da biografia

O DNA é uma estrutura composta por uma hélice de fita dupla. As duas fitas são conectadas por pontes de hidrogênio e cada extremidade da ligação possui uma nucleobase. As possíveis nucleobases do DNA são A denina, C itosina, G uanina e T umina, de forma que A sempre pareia com T e C sempre pareia com G



3.1. fitas de DNA

Para este experimento fictício, consideramos a fita principal como a fita molde, *ou seja*, a fita que é usada para transcrever o RNA. Também para este experimento, os segmentos da fita principal apresentados sempre começarão com a sequência de nucleobases: CAT, então é uma tarefa simples diferenciar a fita molde da fita complementar ou computar uma da outra apenas invertendo os pares de nucleobases.



4. Codificação de DNA

Você deve esperar segmentos de fita de DNA codificados em formatos binários e de string:

4.1. Formato binário

É o formato curto usado para melhor transmissão de dados e desempenho de armazenamento. Neste formato, as nucleobases são codificadas em matrizes de 2 bits:

```
A: 0b00      C: 0b01
T: 0b11      G: 0b10
```

4.2. Formato de string

Usado para uma melhor compreensão humana. Neste formato, as nucleobases são codificadas como seu char: "CATCGTCAGGACTCAGTCCATCTTA ACTACTAAACTC..."

4.3. Exemplo de codificação

Exemplo de codificação de string para formato binário:

```
String:    "CATCGTCAGGAC"
Bits:      0b010011011011010010100001
Byte[]:    [0x4D, 0xB4, 0xA1] // notice the bits to byte conversion is
Big-Endian
Base64:    "TbSh"
```

5. Operações

5.1. Solicitar um trabalho

Base address: <https://gene.lacuna.cc/>

```

'[GET] /api/dna/jobs'
Header
  Authorization = 'Bearer <AccessToken>' // <AccessToken> aquired on 2.2

Response
{
  job?: {
    // Job id
    id: string,

    // Operation types ['DecodeStrand', 'EncodeStrand', 'CheckGene']
    type: string,

    // Strand in String format. Non-null when operation type 'EncodeStrand'
    strand?: string,

    // Strand in the Binary format Base64 encoded. Non-null when operation
    types 'DecodeStrand' and 'CheckGene'
    strandEncoded?: string,

    // A gene segment in the Binary format Base64 encoded. Non-null when
    operation type 'CheckGene'
    geneEncoded?: string
  },
  code: string, // ['Success', 'Error', 'Unauthorized']
  message?: string
}

```

Se tudo estiver OK, você receberá um objeto de trabalho com o trabalho `id`, a operação `type` e os parâmetros de operação que você poderá resolver da seguinte maneira.

5.2. Operação de decodificação de fita

Se você receber uma 'DecodeStrand' operação, o trabalho é pegar o `strandEncoded` parâmetro, que é uma string Base64 da vertente no formato Binário, e decodificá-lo para o formato String de acordo com a sessão 4 .

Para esta operação deverá enviar a resposta para:

```

'[POST] /api/dna/jobs/{id}/decode'
URL Parameters
  id // The Job id

Header
  Authorization = 'Bearer <AccessToken>' // <AccessToken> aquired on 2.2

Request body
{

```

```

    // Decoded strand in String format
    strand: string,
}

Response
{
  code: string, // ['Success', 'Error', 'Fail', 'Unauthorized']
  message?: string
}

```

5.3. Codificar operação de fita

Se você receber uma 'EncodeStrand' operação, o trabalho é pegar o strand parâmetro, que é a vertente no formato String, e codificá-lo para o formato Binário de acordo com a sessão 4 .

Para esta operação deverá enviar a resposta para:

```

'[POST] /api/dna/jobs/{id}/encode'
URL Parameters
  id // The Job id

Header
  Authorization = 'Bearer <AccessToken>' // <AccessToken> aquired on 2.2

Request body
{
  // Encoded strand in Binary format Base64
  strandEncoded: string,
}

Response
{
  code: string, // ['Success', 'Error', 'Fail', 'Unauthorized']
  message?: string
}

```

5.4. Verifique a operação do gene

Se você receber uma 'CheckGene' operação, o trabalho é dizer se um determinado gene está ou não ativado na cadeia de DNA recuperada. Ambos os genes e fitas de DNA são recuperados em formatos binários.

Para este experimento, um gene é considerado ativado se mais de 50% de seu conteúdo estiver presente na fita molde do DNA. Ex:

Gene:

TACCGCTTCA**TAAACCGCTAGACTGCATGATCG**GGT

Cadeia modelo de DNA:

CATCTCAGTCCTACTAACTCGCGAAGCTCATACTAGCTAC**TAAACCGCTAGACTGCATGATCG**CATAGCTAGCTACGCT

No exemplo acima, mais de 50% do gene (~63% do gene) está presente na fita molde, portanto, neste caso, o gene é considerado ativado .

OBSERVAÇÃO: Observe que a comparação do gene deve ser aplicada sobre a fita molde do DNA . Portanto, você precisa verificar de acordo com a sessão 3.1 se a fita recuperada é o modelo ou a complementar e computar uma à outra, se necessário, antes de procurar a presença de segmentos gênicos.

Para esta operação deverá enviar a resposta para:

```
'[POST] /api/dna/jobs/{id}/gene'
```

URL Parameters

```
id // The Job id
```

Header

```
Authorization = 'Bearer <AccessToken>' // <AccessToken> aquired on 2.2
```

Request body

```
{
  // Whether or not the gene is activated in the template strand
  isActivated: boolean,
}
```

Response

```
{
  code: string, // ['Success', 'Error', 'Fail', 'Unauthorized']
  message?: string
}
```

6. Comentários finais

Não há conhecimento que não seja poder

Tenha um teste divertido!!

