

# Caderno Computacional 1/6: Instalação de R, RStudio e o Formato R Markdown

Gepoliano Chaves, Ph. D.

Novembro, 2020

## Contents

<b>Tabela de Conteúdo do Curso</b>	<b>1</b>
<b>1) Introdução</b>	<b>2</b>
<b>2) Conhecimentos de Programação</b>	<b>2</b>
<b>3) Instale o RStudio no Windows</b>	<b>4</b>
3.1) Vídeo de Apresentação do R Markdown . . . . .	6
3.2) Usando R Markdown . . . . .	6
3.3) Tutorial R Markdown em inglês . . . . .	7
<b>4) Instalação de outros programas requeridos ao longo do curso</b>	<b>7</b>
4.1) Lista dos programas utilizados . . . . .	7
4.2) Comandos para instalação dos programas . . . . .	7
4.2.1) Programas instalados em R . . . . .	7
4.2.2) Programas instalados em bash . . . . .	8

## Tabela de Conteúdo do Curso

Table 1: Conteúdo do Curso

Atividade	Descrição	Encontro
1	Semana 1: Apresentação, R, RStudio e RMarkdown	22/03/2021
2	Semana 1: Auxílio à Instalação do R e RStudio	24/03/2021
3	Semana 1: Descrição das Atividades Computacionais	25/03/2021
4	Semana 2: Comandos Usados no R; Livrarias; Plotagem no R	29/03/2021
5	Semana 2: Associação Biológica	31/03/2021
6	Semana 2: Auxílio à sintaxe do R para Associação Biológica	02/04/2021
7	Semana 3: Visualização da Associação Biológica: Manhattan Plot	05/04/2021
8	Semana 3: Agrupamento Hierárquico e Heatmap	07/04/2021
9	Semana 3: Frequência Alélica e Visualização de Frequência Alélica	09/04/2021
10	Semana 4: Dúvidas e auxílio computacional	12/04/2021
11	Semana 4: Estudos Futuros	14/04/2021

# 1) Introdução

R Markdown é um formato de arquivo usado no RStudio que possibilita interação entre a linguagem humana escrita e a linguagem de programação, utilizada pela máquina para executar uma tarefa. O Markdown possibilita a interação da linguagem escrita com vários tipos de linguagem de programação, como R, Python e Bash (Linux). Neste documento, aponto links para vídeos que ensinam a instalar e utilizar o RStudio, assim como diversas utilidades e sintaxes usadas no R Markdown.

## 2) Conhecimentos de Programação

Este curso não assume familiarização do estudante com conceitos de programação computacional. No entanto, por tratar de tópicos vastos como genética, genômica, biologia computacional e sequenciamento de DNA, o conteúdo apresentado pode mostrar-se complexo e desgastante para alunos não familiarizados com estas áreas. Além disso, a própria instalação de pacotes de programas em R e bash e a utilização correta da sintaxe de cada linguagem, pode ser um desafio em si mesmo. Precisamos portanto, alertar os estudantes, da necessidade de abstração no que se refere à necessidade da escrita de códigos de programação. Um exercício muito útil é escrever a ideia por trás de um comando antes de buscar a forma apropriada de execução do mesmo. Discorrer sobre a execução de um código é necessário tanto antes da execução do comando, quanto após a finalização do mesmo, como forma de aprendizado. Em cursos acadêmicos de Biologia Computacional por exemplo, a escrita do código requer uma etapa chamada formulação de pseudo-código, em que o comando é desenhado em linguagem comum para leitura humana, antes de desenhá-lo no código para execução pela máquina (Figura 1).

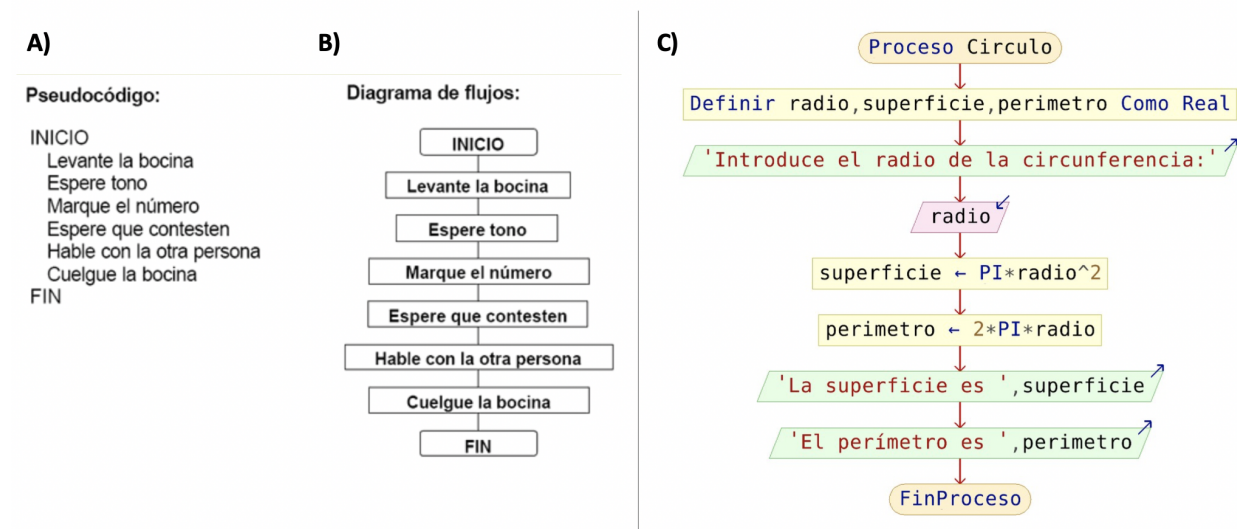


Figure 1: Pseudo-código desenhado para leitura em linguagem humana. A) Uma atividade cotidiana escrita em pseudo-código; B) O pseudo-código em Diagrama de Fluxo; C) Pseudocódigo de dois processos matemáticos: o cálculo da superfície e do perímetro do círculo.

Referências usadas na Figura 1:

<https://openwebinars.net/blog/que-es-pseudocodigo/>

<https://sites.google.com/site/info080910/lenguajes-de-programacion>

Por não requerer proficiência em nenhuma linguagem de programação, a estratégia deste curso será a busca de comandos de interesse em navegador de internet. Esta busca representa um esforço por parte do aluno

afim de que a informação proveniente da internet seja usada de forma correta. Para ilustrar esta idéia, da busca pelo comando apropriado para execução de uma tarefa, imagine que eu te apresente um arquivo de texto chamado “Arquivo.txt”. Agora, peço que você abra o arquivo, e leia para mim, o conteúdo do mesmo. No Windows, esta tarefa é bem básica para usuários intermediários, visto que o usuário apenas necessita “clicar” no arquivo para ver o conteúdo. Aliás, em muitos computadores que usam em sistema operacional diferente do Windows, e mesmo muitas máquinas Windows, é possível visualizar o conteúdo do arquivo sem clicar para abri-lo, como no exemplo da Figura 2.

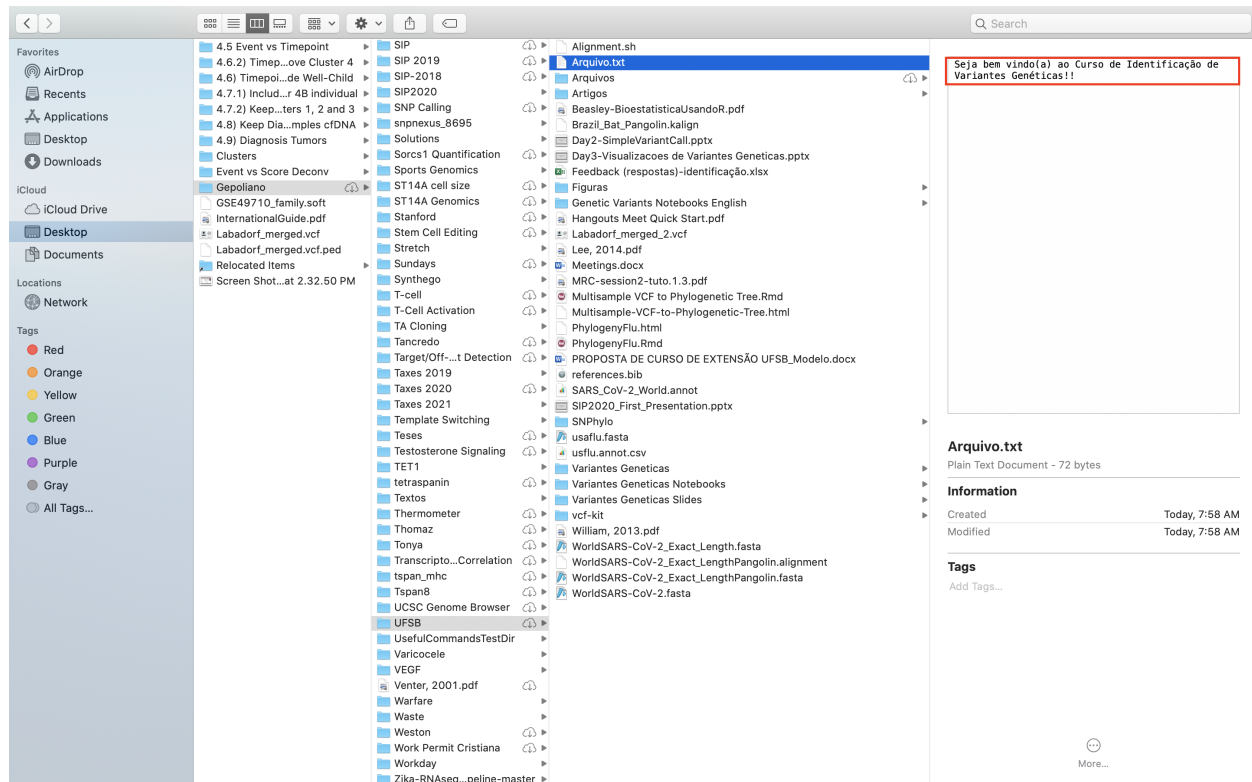


Figure 2: Lendo o conteúdo de um arquivo de texto. Em muitos sistemas operacionais, esta tarefa é facilmente executada simplesmente clicando no arquivo, sem necessidade de abri-lo, como ilustrado pela caixa vermelha.

No entanto, embora possamos processar muitos arquivos da mesma forma que estamos acostumados, como usuários comuns do Windows, nos Cadernos de Biologia Computacional deste curso, este tipo de processamento não será suficiente. Visto que queremos abrir o arquivo e ler o conteúdo, sugiro que você vá ao Google, ou outro navegador de sua preferência, e busque por “linux abrir arquivo de texto” ou “bash abrir arquivo de texto”. Não tenho certeza de que resposta você encontrará. Eu poderia mostrar a você a resposta que eu encontrei. Porém, desejo apresentar a linha de comando do Linux usando o comando `head`. Honestamente, os resultados da minha busca não me mostraram o comando `head`. No meu navegador de internet, então, digitei “linux comando head português”, e sugiro a você que faça o mesmo. Na seguinte página, que me pareceu muito útil, encontramos a seção “Exibição ou Impressão de Arquivos”:

<https://www.devmedia.com.br/comandos-importantes-linux/23893>

Na página acima, encontro a informação que desejava: “Head: Mostra as primeiras linhas de um arquivo, como por exemplo com `head -10 a.txt`, ou usado como filtro para mostrar apenas os primeiros x resultados de outro comando”. Numa visualização da página acima, mostro a descrição do comando `head`, que vamos usar a seguir (Figura 3).



Neste curso, os comandos para identificação de variantes genéticas foram adquiridos pela leitura de artigos científicos, pacotes de programação de R e bash e tutoriais de utilização destes pacotes. No entanto, também foi necessário buscar comandos para processamento de arquivos, os quais não estavam presentes nestes pacotes. Esta busca por outros comandos sempre foi motivada pela pergunta ou hipótese científica. Agora, nossa pergunta é como podemos visualizar nosso Arquivo.txt usando o comando head em bash. Para responder a esta pergunta, simplesmente cliço na aba “insert” do Markdown, inserindo um *chunk* para ser escrito em bash (Figura 3).

Uma vez que o chunk foi inserido, digitamos o comando head, e depois, o caminho correto para a pasta contendo nosso arquivo de interesse, para executar a tarefa de ler o conteúdo do arquivo:

```
head ~/Desktop/Gepoliano/UFSB/Arquivo.txt
```

Assim, iniciamos a ambientação de estudantes com a instalação de programas a serem usados ao longo do curso, começando pelo RStudio.

### 3) Instale o RStudio no Windows

Não costumo usar o sistema operacional Windows atualmente. Embora tenha usado o RStudio no Windows no passado, nunca usei o R Markdown no Windows. Hoje em dia uso O RStudio em uma máquina MacBook Pro, onde uso a linha de comando diretamente no Terminal. Pelo fato de o Mac já vir com o terminal instalado, sempre achei o Mac mais conveniente do que o Windows pois no Windows, devido a minhas necessidades de usar o Terminal, precisava instalar algum programa para interagir com o Linux localmente ou em algum servidor remoto. Precisei desenvolver habilidades de interação com o Linux ao usar um supercomputador na Universidade da Califórnia em Santa Cruz, durante o doutorado. O supercomputador, chamado Hummingbird, possui capacidade de armazenamento e programas instalados para processamento de dados genômicos usados no laboratório.

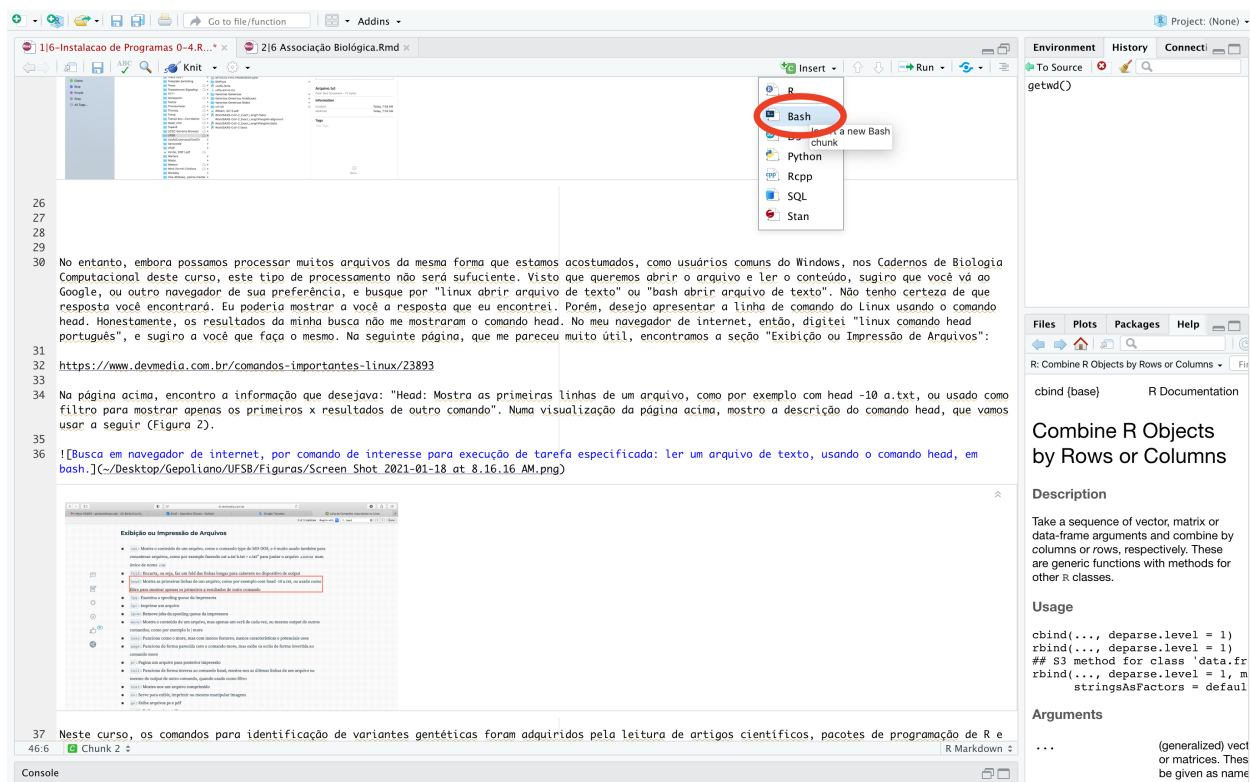


Figure 3: Inserção de *chunk* para ser escrito em bash. A inserção do chunk pode ser feita clicando-se na aba *insert* (acima da bola circular em vermelho) e escolhendo-se a opção bash.

Neste curso, nossa interação será através do RStudio. Antes de instalar o RStudio, porém, precisamos instalar o R.

No vídeo abaixo, Mercei Santos ensina como instalar o R, necessário para posterior instalação do RStudio.

<https://www.youtube.com/watch?v=G03TwotgPQ8>

A seguir, ele ensina como instalar o RStudio, que hospeda o formato R Markdown, no windows:

<https://www.youtube.com/watch?v=aeK6-kPhofk>

### 3.1) Vídeo de Apresentação do R Markdown

Gostaria que ficasse claro no presente Caderno de Biologia Computacional, que uma importante utilidade do R Markdown é a fácil interação entre a linguagem escrita e o apontamento para *links* da web que possibilitem ao leitor interação com o conteúdo que explique meios de utilização do R Markdown. Em diversos momentos na escrita dos Cadernos de Biologia Computacional, precisei e ainda preciso, da utilização de um código que eu não sabia como escrever. Nestes momentos, sempre tenho de passar alguns minutos, às vezes horas, navegando a internet para encontrar o comando numa linguagem que me sirva. Talvez no futuro, o leitor seja levado a ter uma experiência similar, onde tenha de procurar pela sintaxe correta para execução de um comando, ou talvez tenha de usar o comando puro e simples. Nestas horas, o conceito da palavra reprodutibilidade deve ser enfatizado. É muito importante deixar anotado, o site da web que visitou para encontrar o comando, assim como um esboço da necessidade que o pesquisador apresentava, possivelmente declarado antes da referência do site. Acho esta uma característica importante do Markdown, que é a possibilidade de escrever suas notas, em interação com o link do site onde encontrou um comando importante. Desta forma, o pesquisador terá uma boa reprodutibilidade no fluxo de idéias, hipóteses e pensamentos que permeiam suas análises, permitindo que os dados sejam melhor analisados que se o pesquisador dependesse de proficiência pura dos comandos que utilizasse. No próximo vídeo, Kaíque Alves faz uma breve introdução ao uso do R Markdown. Por exemplo, ele explica a idéia de *chunks* ou pedaços de código. Cada *chunk* pode ser escrito em uma linguagem específica, por exemplo R ou Python, e então, executado.

<https://www.youtube.com/watch?v=gF7cNGR3a4Y>

Esta característica do R Markdown, de oferecer a interação de códigos escritos em diferentes linguagens, é muito vantajosa para o pesquisador que utiliza múltiplas línguas de programação. Por exemplo, um conjunto de dados pode ser processado por um comando em bash, exportado, e usado como *input* por um pacote em R.

### 3.2) Usando R Markdown

O canal Zurubabel traz uma introdução geral ao R Markdown, ensinando sobre a produção, formatação e download do documento, incluindo como realça o texto em negrito e itálico. O canal também fornece uma noção de como formatar a estrutura de divisão do texto do arquivo, dando ao pesquisador uma visão dos níveis hierárquicos presentes no arquivo em produção. O interessante deste Canal também é sua interação com o Github, uma plataforma onde códigos são disponibilizados e explorados no vídeo 10.

<https://www.youtube.com/watch?v=mhevC54PWAs&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=1> [https://www.youtube.com/watch?v=\\_4xurljld0Y&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=2](https://www.youtube.com/watch?v=_4xurljld0Y&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=2) <https://www.youtube.com/watch?v=rheCuWorMBs&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=3> <https://www.youtube.com/watch?v=C617MES0YT4&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=4> <https://www.youtube.com/watch?v=JR7QttAJwdc&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=5> <https://www.youtube.com/watch?v=kFlfNCyHGb4&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=6> <https://www.youtube.com/watch?v=kFlfNCyHGb4&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=7> <https://www.youtube.com/watch?v=kFlfNCyHGb4&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=8> <https://www.youtube.com/watch?v=kFlfNCyHGb4&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=9>

index=10 <https://www.youtube.com/watch?v=GCWdR4O-YJo&list=PL4OAe-tL47sZJVgc80pJazcy4iI59k5yy&index=11>

### 3.3) Tutorial R Markdown em inglês

Este tutorial foi o primeiro que utilizei para aprender sobre a sintaxe usada no R Markdown. Como foi o primeiro que usei, achei que poderia ser útil do ponto de vista de que eu estar de alguma forma mais familiarizado com este material. Porém este vídeo está em inglês e se você não quiser assistir o vídeo assim, o material disponibilizado acima deverá ser suficiente em termos de provisão de informações necessárias sobre R Markdown para o propósito deste curso.

<https://www.youtube.com/watch?v=tKUufzpoHDE&t=991s>

## 4) Instalação de outros programas requeridos ao longo do curso

Seria ideal já ensinar a instalação de bgzip, tabix, BWA, vcftools, bcftools, PLINK, QQman entre outros, para facilitar a ambientação do estudante com estes programas.

### 4.1) Lista dos programas utilizados

- bwa
- bgzip
- tabix
- vcftools
- bcftools
- PLINK
- QQman
- SRA-Tools
- GATK
- Picard

### 4.2) Comandos para instalação dos programas

Neste curso, usaremos basicamente comandos escritos em R e em bash. Nas duas próximas seções, instalamos alguns dos programas a serem usados ao longo deste curso.

#### 4.2.1) Programas instalados em R

Por vezes é necessário incluir o comando `eval = F`, para que o PDF possa ser plotado devidamente.

```
install.packages("qqman")
```

#### 4.2.2) Programas instalados em bash

R Markdown não aceita executar o programa da mesma forma como o programa é executado na linha de comando. No Markdown, o seguinte comando para instalar um programa usando Anaconda, simplesmente não funciona, então preciso descobrir como consertar isto.

```
conda install -c bioconda bwa
```

Porém, ao prover o caminho completo para a pasta de instalação de Anaconda, o comando pode ser executado, e os programas podem ser instalados:

```
/Users/gepolianochaves/anaconda3/condabin/conda install -c bioconda bwa  
/Users/gepolianochaves/anaconda3/bin/bwa
```