Maira Neves

Mestre em Ciências (ênfase: Biologia da relação patógeno-hospedeiro) formada pelo Departamento de Parasitologia do Instituto de Ciências Biomédicas da Universidade de São Paulo (2014), e formada em Bacharelado e Licenciatura em Ciências Biológicas pela Universidade de São Paulo (2011;2014).

Aluna de doutorado do programa de Bioinformática, orientada pelo Prof. Carlos Frederico Martins Menck.

Título do projeto: Avaliação transcricional de células pluripotentes induzidas e neuro-precursoras provenientes de pacientes com síndrome de Cockayne após indução de dano ao DNA.

---

exec

Plano A: Lista de compras de supermercado planejada e cumulativa

Contextualização

A vida moderna traz muitas dificuldades, como a escolha de alimentos no supermercado. São tantas opções, que gente comilona sempre acaba comprando mais comida do que precisa, e deixando estragar um tanto da comida que comprou por falta de planejamento. Com isso veio a motivação de criar uma função que pudesse auxiliar o momento das compras de supermercado, para que a gente não se perca nas escolhas. A ideia é que dado um número de refeições que se vá fazer em casa, a função possa escolher os ingredientes necessários para o preparo das refeições daquele período, de acordo com a classe da refeição (café da manhã ou da tarde, refeição principal, refeição leve ou pé na jaca). A função já irá possuir um repertório de receitas para cada uma das categorias, mas deverá aceitar novas receitas. Além disso, a função oferecerá a possibilidade de preparar uma refeição principal para mais de uma pessoa, calculando as quantidades adequadas. A função também vai ajudar a evitar o desperdício, então deverá aproveitar os ingredientes que sobraram em casa.

Entrada

- Número de cafés (manhã ou tarde) para o período. Padrão: 0.
- Número de refeições principais. Padrão: 0.
- Número de refeições leves. Padrão: 0.
- Número de pé na jaca. Padrão: 0.
- Número de refeições múltiplas. Padrão: 0.
- Categoria da receita adicional: “cafe”, “principal”, “leve”, “jaca”.
- Receita adicional. Variável declarada anteriormente contendo dataframe, sendo coluna 1: nome do ingrediente, coluna 2: quantidade, coluna 3: unidade de medida. Padrão: NA.
- Sobras da última compra. Variável declarada anteriormente contendo dataframe, sendo coluna 1: nome do ingrediente, coluna 2: quantidade, coluna 3: unidade de medida. Padrão: NA.

Planejamento da função

1. Verificar se os primeiros 5 parâmetros são numéricos, e se existirem tabelas declaradas para os dois últimos parâmetros, se contém 3 colunas e a segunda coluna é numérica.

2. Escrever dataframes contendo receitas e armazenar as receitas na lista apropriada. Total de 4 listas, uma para cada categoria de refeição.
3. Escrever lista vazia que servirá para armazenar a lista de receitas.
4. Escrever dataframe vazio que servirá para armazenar a lista de supermercado.
5. Se receita adicional e categoria foram declaradas, adicionar esta a lista de receitas apropriada.
6. Se sobras foi declarado, escrever dataframe vazio que servirá para retornar as sobras que não serão utilizadas.
7. Se sobras foi declarado, buscar receitas que contenham os ingredientes declarados, até o número máximo de receitas declaradas de cada tipo.
8. Adicionar cada receita utilizada na lista de receitas.
9. Caso sobrem ingredientes da lista sobras (pois não há mais possibilidade de atribuir receitas pelo número de refeições a serem planejadas), adicionar os ingredientes e quantidades excedentes ao dataframe de sobras.
10. Sortear receitas para as refeições que não foram planejadas para usar sobras.
11. Adicionar essas receitas à lista de receitas.
12. Unir todos os ingredientes das receitas na lista no dataframe de lista de supermercado.
13. Subtrair as quantidades de ingredientes presentes na sobra.
14. Retornar sobra, lista de supermercado e lista de receitas.

Saída

Uma lista contendo os dataframes das receitas utilizadas, uma lista de supermercado, uma lista de sobras.

Comentários Andre Chalom

Alô, Maira!

Ótima proposta, com bom humor e aplicação universal!

Sugestões:

1. Me parece uma proposta bastante trabalhosa para o tempo que vocês tem. Sugiro reduzir um pouco a complexidade (tirando por exemplo a “receita adicional” e a categoria “refeição leve”, já está de bom tamanho sem isso) da proposta… melhor se comprometer com menos, e se der tempo fazer tudo o que vc quer.
2. Seria interessante que os “livros” de receita pudessem ser fornecidos pelo usuário, ao invés de construídos dentro da função. Vc pode mandar um “livro” de receitas para cada categoria nos exemplos da função

Resposta Maira

Oi André,

Muito obrigada pelos comentários.

Decidi seguir com a proposta 1. Acho que me diverti mais escrevendo esta, e tenho a sensação de que vou usar mais repertório de código do R para escrevê-la.

Obrigada mais uma vez pelas sugestões, vou acatar suas sugestões a essa proposta, acho que realmente vai ser mais simples assim.

Plano B: Transformação de FPKM em TPM e exploração dos dados de transcriptoma

Contextualização

A maior parte dos dados de expressão de genes ou transcritos em um transcriptoma são representados pela métrica RPKM ou FPKM (reads/fragments per kilobase of exon per million reads mapped)[1]. No entanto, a soma dos FPKMs de cada amostra de um mesmo experimento é diferente, e portanto comparações diretas entre valores de FPKM de amostras de um mesmo experimento não são confiáveis[2]. Isso é um grande problema, já que todo pesquisador quer poder comparar os dados obtidos na sua condição experimental com os dados do controle, e confiar nessa comparação.

O mesmo artigo citado anteriormente[2] propõe uma métrica alternativa para resolver este problema, e permitir a comparação direta entre amostras de um mesmo experimento. Esta métrica é o TPM (transcripts per million), e se relaciona com o FPKM da seguinte maneira:

sendo i o transcrito de interesse, e j, cada um dos transcritos daquela amostra.

A proposta desta função é calcular os TPMs baseados em dados expressos em FPKM, criar um gráfico de distribuição de TPMs na forma de boxplot (frequentemente utilizado para monitorar a se a diversidade de abundância dos transcritos muda entre amostras), e boxplot representando os TPMs de um mesmo gene ou transcrito nas amostras de acordo com o seu tratamento.

Entrada

- dataframe previamente declarado como variável, contendo os FPKMs, apresentando nas linhas cada um dos genes ou transcritos, e nas colunas cada uma das amostras. O nome das amostras deverá iniciar com a condição a qual pertence aquela amostra. Exemplo: tratamento-1, tratamento-2, controle-1, controle-2.
- genes ou transcritos de interesse para inspeção nos boxplots a serem gerados, apresentados como lista de strings.

Planejamento da função

1. Verificar parâmetros.
2. Verificar se cada uma das strings integrantes da lista do segundo parâmetro está presente como nome de linha do dataframe.
3. Criar dataframe vazio para incluir os TPMs calculados.
4. Realizar a conversão de FPKM para TPM para todas as linhas de cada amostra e armazenar no dataframe criado anteriormente.
5. Desenhar um boxplot de distribuição de TPM por amostra, colorindo a amostra de acordo com a condição.
6. Criar um loop para percorrer cada um dos itens da lista declarada no segundo parâmetro.
7. Desenhar um boxplot para cada um dos itens, representando a distribuição de TPMs daquele item por condição.

Saída

Uma tabela contendo os TPMs de cada gene ou transcrito de cada amostra, um boxplot de

distribuição de TPM por amostra, e um boxplot para TPM de cada gene ou transcrito declarado de acordo com a condição.

Comentários Andre Chalom

Outra proposta interessante, com potencial de ser usada por bastante gente. Algumas dúvidas:

1. Quando você diz que “O nome das amostras deverá iniciar com a condição a qual pertence aquela amostra”, você está falando dos **colnames** da entrada de dados? Acho que isso precisa estar mais explícito. Também é importante explicitar que você espera que sejam usados “-” pra separar a condição e o número da amostra (se os nomes forem “tratamento.1” e “tratamento.2” sua função pode reclamar?)
2. Não entendi direito o último passo do planejamento, você pode elaborar melhor o que isso vai fazer?

Trabalho Final

Plano A

Arquivos de exemplo de livros de receitas para a função: cafe.csv jaca.csv refeicao.csv

Link para a página com o código da função Função lista_compras

Link para a página com a ajuda da função Help