Cainã Max Couto-Silva

Biólogo, mestre e doutorando em genética e biologia evolutiva na USP.
Área de concentração: Genética populacional (humana).

Contatos

- ***E-mail:*** cmcouto.silva@gmail.com || cmcouto.silva@usp.br
- ***Celular:*** 11 95242-3158.

Exercícios

exec

Proposta de Trabalho Final

PROPOSTA 1

Função:

```
link.des()
```

Contextualização

A diversidade genética e sinais de seleção natural compreendem fatores importantes para a

compreensão da nossa história evolutiva. Na área da evolução e genética populacional humana, arrays de SNPs (polimorfismos de nucleotídeos únicos) têm sido muito utilizados para detecção de sinais evolutivos sob um cenário neutro.

Nas últimas décadas têm surgido uma variedade de métodos estatísticos para detectar sinais de seleção natural em datasets genômicos. Dentre elas, destaca-se aqui os métodos por desequilíbrio de ligação, visando identificar haplótipos com homozigose extendida, uma vez que um alelo importante sob potencial seleção aumenta rapidamente de frequência e − consequentemente − traz consigo também seus alelos vizinhos, promovendo uma extensão de homozigose de modo que a recombinação genética não é capaz reestabelecer a heterogeneidade esperada na população. Tal método consiste em uma poderosa ferramenta para detecção de uma seleção positiva intensa em um período relativamente curto de tempo. Alguns exemplos de testes estatísticos cujas construções adotam o princípio descrito acima são: EHH, iHS e XP-EHH, e estão disponíveis em um pacote do R denominado “rehh”. Eu, particularmente, utilizei este pacote recentemente e, para análises das minhas populações de estudo, cromossomos/genes, SNPs e alelos (ancestrais ou derivados), tenho escrito scripts com aproximadamente 3000 (três mil) linhas, o que gera um trabalho relativamente árduo. Assim, proponho aqui a construção de uma função que, em uma chamada e seus parâmetros, analise e gere os gráficos e as tabelas para todas as populações de interesse, cromossomos e SNPs de interesse, bem como seus respectivos alelos.

Argumentos de entrada

A ideia é que o usuário insira objetos-vetores (principalmente) contendo um conjunto de valores para cada argumento, por exemplo: - Vetor com os nomes das populações (nomes das populações); - Vetor com os cromossomos a serem analisados (vetor numérico); - Lista com as identificações dos SNPs de interesse e seu respectivo cromossomo;

E por trás disso ter-se-á diversos loops, leitura de dados, criação de pastas, escrita de tabelas e salvamento de plots/figuras. Dentro da função muitos objetos criados serão imediatamente utilizados como entrada para outros parâmetros do pacote rehh, cujas saídas serão utilizadas para outros parâmetros, e assim por diante para enfim gerar todas tabelas e gráficos a serem armazenados de forma organizada nas pastas criadas (por isso iria sugerir utilizar a função em uma pasta vazia, apenas com os arquivos iniciais necessários para os argumentos em uma pasta com algum nome específico, e as demais pastas terão nomenclatura e organização padrão conforme o input do usuário).

Arquivos de saída

Criação dos objetos do pacote rehh, pastas com tabelas e gráficos organizados por teste/população. Exemplos:

- EHH (diagramas de bifurcação e gráficos lineares comparando alelos ancestrais/derivados, etc)
- iHS (tabelas com valores significantes segundo a literatura e representação em plots)
- XP-EHH (idem iHS para comparação entre duas populações com base também no teste de Fst).

Resumindo, a função visa automatizar um processo que geraria um script imenso (e que estará internamente na função) para que o usuário possa digitar apenas uma linha e, com a função, gerar todos os testes disponíveis no pacote rehh para seus dados de interesse. O viés é que removeria um

pouco da liberdade do usuário, um viés que pode ser amenizado com a adição de mais parâmetros (como T ou F para alguns testes, determinação de um nome padrão para os gráficos segundo arquivos de entrada, etc).

Acredito que ficaria uma função internamente grande e com muitas manipulações de texto, meio trabalhosa, porém factível e útil para futuras análises utilizando tal pacote, e que pode ser melhor trabalhada posteriormente.

> Olá Cainã,
>
> Pelo que entendi, o propósito principal de sua função é automatizar um processo de análise e visualização de dados que é muito longo de ser executado, especificamente no contexto da sua área de pesquisa. Embora um dos objetivos de criar funções no R seja esse que você quer atingir com essa proposta, considero que para o exercício do curso não é tão bom assim. Uma das particularidades do R é que o usuário digita o código e pode saber o que cada linha faz, contrário à maioria de pacotes estatísticos. Assim, sua função pode acabar escondendo o processo que você quer executar, o que é 'bom' para você (pense de verdade quais seus ganhos com isso) mas não seria para outros usuários.
>
> Por outro lado, e voltando aos propósitos e instruções para o trabalho final, evite focar sua proposta de função no contexto da sua pesquisa e mais no que a função faria em contexto geral. Recomendo focar na parte de análise ou visualização dos dados e não nas duas, no caso de você seguir em frente com essa sua proposta. Não estão claros os objetos de entrada nem de saída da função. Tente focar num tipo de análise particular e explicar melhor quais seriam os tipos de dados com que a função lidaria, os argumentos e valores dos mesmos, e os objetos de saída. Além disso, as funções do trabalho final devem estar baseadas nos pacotes básicos. — *Gustavo Agudelo 2017/06/02 18:18*

> Olá Gustavo, tudo bem? Desculpe-me pela demora em responder, espero que não seja muito tarde. Como a resposta será explanativa e um pouco longa, deixarei abaixo a mesma proprosta reescrita visando fornecer uma melhor compreensão.
>
> Sobre esta primeira proposta, caso não possa utilizar o pacote que mencionei, a função inteira perderia por completo seu sentido, visto que elaborei a proposta inteira em cima dela. E como não tem nenhuma regra quanto ao uso restrito dos pacotes básicos do R na página do Trabalho

> Final, eu realmente imaginei que poderia utilizar nestes casos. Mas peço para reconsiderar (se puder), pois não é uma função que irei utilizar a fim de facilitar meu trabalho, e considero que além de uma função muito útil para quem iria utilizar tal pacote, um bom exercício de programação também. Tentarei explicar melhor na proposta reestruturada, e espero teu respaldo para ver se posso dar continuidade com a primeira proposta.
>
> Att., — *Cainã Max Couto da Silva 2017/06/12 15:00*

Referências:

https://www.ncbi.nlm.nih.gov/pubmed/24274750

https://cran.r-project.org/web/packages/rehh/index.html

PROPOSTA 2

Função para manipulação de dados populacionais, para extração de subpopulações e comparações quantitativas (gráficas), independente do tipo de arquivo de entrada, podendo ser um vetor, matriz, dataframe, lista ou array, bastando o usuário citar qual a classe do arquivo inserido e, dependendo da classe, inserir os argumentos respectivos aos parâmetros da função (para isso terá o help) para especificar aonde estão as identificações dos indivíduos. Por exemplo, em uma entrada como dataframe, o usuário indicaria qual a linha ou coluna que se encontra os indivíduos e, com base no segundo arquivo de entrada que deverá oferecer uma lista de relação população/indivíduos, a função possibilitaria a extração de populações específicas (e seus respectivos dados) e também gráficos e sumários para comparação entre as populações.

Como opções alternativas, apresento mais duas sub-opções:

* Função exploratória: fazer uma função que faça testes e gere gráficos sobre algum conjunto de dados obtidos por uma lista (dataset) que iria criar previamente oriundo da internet (por exemplo, sobre tempo gasto em séries e correlações entre as mesmas);

* Função com algoritmos de previsão: dado parâmetros indicados pelo usuário e conhecimentos prévios disponíveis na literatura científica, prever um resultado. Por exemplo, sabe-se n variáveis influencia x (valor, percentual por exemplo) na variável y, e então o usuário poderá brincar com diferentes valores das variáveis n, e a função calcularia o relativo do já conhecido em x para n a fim de determinar y. Uma ideia seria fornecer argumentos para variáveis com influência na neurogênese, i.e. formação de novos neurônios, na fase adulta, como álcool, atividade física e afins, e o quanto afetaria a neurogênese (y). Evidentemente ter-se-ia que analisar diversos outros fatores, e não apenas consumo, como intensidade, tempo, associação e etc., portanto saliento que seria uma

proposta com fins didáticos de programação em R.

> Cainã, dessa vez essas propostas estão ainda mais confusas, pois você não esclarece o que você quer fazer de fato. A função de algoritmos de previsão, pelo que entendi, pretende prever valores de uma variável resposta em função de uma ou mais variáveis preditoras, como seria feito por ajuste de modelos. Não vejo qual o diferencial. Tente focar no objetivo da função e descreva ele claramente, depois descreva os valores de entrada e saída da função. Lembrando que uma boa função no trabalho final deve conter pelo menos um ciclo de for() ou if(). Se tiver alguma dúvida, mande uma mensagem ou poste no fórum. — *Gustavo Agudelo 2017/06/02 18:40*

> Concordo plenamente com tudo o que disseste, Gustavo, e reconheço que ficou mal explicada devido também à preocupação maior com a proposta anterior, e esta acabou saindo meio que no desespero sobre o quê propor. Irei explanar sobre função para manipulação de dados populacionais originalmente proposta na reestruturação desta proposta, logo abaixo. Att., — *Cainã Max Couto da Silva 2017/06/12 15:00*

Proposta 1 − Reestruturada

Primeiramente irei explanar um pouco sobre o pacote rehh, cuja utilização é essencial nesta função, uma vez que a função proposta é inteiramente desenhada para rodar todas as funções deste pacote considerando todos os arquivos de entrada do usuário. Este pacote (rehh) possui um conjunto de funções para análises que visam identificar sinais de seleção natural (positiva) no genoma, ok!? Anteriormente elaborei uma sucinta introdução contextual mencionando os testes de desequilíbrio de ligação (*linkage disequilibrium − LD*) porque os testes deste pacote adotam este princípio. Logo, as funções do pacote ao final proveem testes estatísticos (baseados no LD) e gráficos que são usualmente utilizados em artigos da área e, portanto, não será de minha responsabilidade ou objetivo da função proposta implementar os testes ou criar os gráficos, pois as funções deste pacote já fazem isso. Por fim, qual seria então o objetivo, utilidade e parte prática de programação em R!?

Para responder isso, mesmo mencionando o fundamento do pacote rehh, ainda se faz importante mencionar as principais funções do pacote rehh e seus respectivos arquivos de entrada. Citarei e deixarei resumido aqui na ordem conforme o protocolo:

data2haplohh. Função utilizada para carregar dois arquivos que contém os dados populacionais/genéticos (SNPs), sendo um deles .map, e criar objetos específicos para análises posteriores.
scan_hh. Função que utiliza como entrada os objetos criados pela função "data2haplohh" a fim de calcular os variados testes baseados no linkage disequilibrium para todos os SNPs de cada cromossomo por população.
ihh2ihs. Calcula/computa o iHS utilizando como entrada os objetivos provenientes da função "scan_hh".
ies2xpehh. Calcula/computa o XP-EHH utilizando como entrada os objetivos provenientes da função "scan_hh" (neste caso, compara-se populações).
ihsplot. Retorna os plots dos objetos criados pela função "ihh2ihs".
xpehhplot. Retorna os plots dos objetos criados pela função "ies2xpehh".
calc_ehh. Calcula/computa o EHH de um SNP focal, "core" (vide contextualização), utilizando como entrada os objetos de saída da função "data2haplohh" e um argumento específico com a posição do SNP de interesse (focal) no objeto haplohh, fornecendo além dos resultados quantitativos do teste, também gráficos representativos, considerando ambos os alelos possíveis (ancestral ou derivado).
bifurcation.diagram. Calcula/computa o EHH de um SNP focal, "core" (vide contextualização), utilizando como entrada os objetos de saída da função "data2haplohh" e um argumento específico com a posição do SNP de interesse (focal) no objeto haplohh, porém, diferentemente da função "calc_ehh", solicita argumento para análise de um alelo específico (ancestral ou derivado), gerando um gráfico de bifurcação deste alelo dado sua posição.

Dentre outras funções não cruciais e que portanto não serão abordadas aqui, uma vez que o intuito é apenas esclarecer. É sumariamente importante salientar aqui, entretanto, que cada arquivo populacional é um arquivo separado, com populações específicas, e para cada um destes arquivos com populações específicas, têm-se os arquivos separados por cromossomos específicos. Por exemplo: chr1_africans.vcf, chr1_europeans.vcf, chr2_africans.vcf, e assim por diante. Então entraria a questão dos SNPs de interesse. Vamos supor, por exemplo, que se queira analisar todos os SNPs da rede do gene p53 (a fim de saber se está sob seleção em alguma das populações), ok!? Logo, você terá uma lista com os SNPs de diferentes genes (correlacionados ao TP53) e que por sua vez terão sua localização conforme o gene/cromossomo e que, portanto, cromossomos diferentes/específicos entram em questão. Nas funções "calc_ehh", por exemplo, teria-se que chamar a função N vezes, sendo que o N aqui representa o número de SNPs (isso após achar sua posição no objeto), enquanto que na função "bifurcation.diagram", seria necessário 2N, uma vez que se deve analisar tanto o alelo ancestral quanto o derivado para cada SNP. Agora, pense se houvesse uma função que rodasse todas as funções do pacote rehh em um único comando, considerando: todos os arquivos de cada população para cada cromossomo e a lista de SNPs de interesse, gerando os testes e criando pastas organizadas para salvar tanto os arquivos de texto e imagens das análises gerais, quanto pastas com arquivos/imagens com os dados dos SNPs interesse. E eis o intuito capcioso da função proposta rs.

Vamos ver se consigo deixar de forma mais clara agora:

Função: `link.des()`

Objetivo: Rodar todas as funções (escolhidas pelo usuário) do pacote rehh para todo o dataset de interesse do usuário em um único comando em R

Argumentos de entrada:

obrigatórios

- `pop = <arquivo contendo o nome das populações>`
- `map = <arquivo contendo o a posição dos SNPs>`
- `snps = <lista de SNPs de interesse>`

opcionais

- `chr = 1:22` (padrão, porém poderia analisar também apenas cromossomos específicos)
- `ihh = T` (calcular iHS; padrão é True)
- `xpehh T` (calcular XP-EHH; padrão é True)
- `ihh.plot = T` (plot do resultado iHS; padrão é True)
- `xpehh.plot = T` (plot do resultado XP-EHH; padrão é True)
- `ehh = T` (calcula e faz o plot do EHH para todos os SNPs/chr da lista fornecida; padrão é True)
- `bifurcation.diagram = T` (plot do EHH para ambos os alelos de todos os SNPs/cromossomo da lista fornecida; padrão é True)

Alterando-se os parâmetros para F nos argumentos opcionais, a função correspondente do pacote rehh não será executada)

Objetos e arquivos de saída:

- Pastas com os nomes de cada população, contendo todos os arquivos-resultado dos testes para a população em formato txt, bem como os plots representativos, considerando os SNPs de interesse da pesquisa (lista fornecida);
- Pasta contendo o iHS de todos os cromossomos juntos para cada população;
- Pasta contendo o XP-EHH de conjunto de cromossomos para todas as combinações dois-a-dois possíveis entre as populações.

Justificativa: Como os arquivos e argumentos de entrada são muito variáveis, levando-se em consideração todas as populações, todos os cromossomos por população, e todos SNPs por cromossomo, muitas chamadas de funções pelo pacote rehh far-se-iam necessárias, principalmente aquelas com necessidade de especificar posição de SNP e modalidade dos alelos. Ademais, muita cautela com a nomenclatura de todos objetos criados e seus constantes (re)usos e novas nomenclaturas são também requisitadas. A função proposta visa promover uma enorme economia de tempo e maior segurança contra possíveis erros de digitação referentes aos objetos e parâmetros das funções. O conhecimento prévio de como trabalha as funções e logística do pacote rehh serão de todo modo necessárias.

Desafios no quesito programação: Para seguir o protocolo que deveria ser executado manualmente pelas funções do pacote rehh, ter-se-ia que rodar função por função, pegando-se os objetos de saída de uma e rodando na outra, criando-se novos objetos, e assim por diante. Assim, para automatizar tudo isso, diversos loops, while e for com if, serão utilizados, o que proporcionará uma boa experiência prática com loops e condicionais em R. Além disso, a automatização de um protocolo com muitos loops, condicionais e criação/utilização consecutiva de objetos, irá exigir conhecimento e boas práticas em manipulação de arquivos, textos, split, imagens/plots, dos devices em R, e muita indexação, inclusive porque será necessário também a criação de pastas e direcionamento de arquivos. Devido à isso alguns usuários do pacote variam entre as funções deste pacote e o uso do shell/bash/Perl para manipular arquivos texto e gerar outros arquivos texto que serão utilizados para rodar os testes para SNPs de interesse (parte complicada de entender sem visualização prática, I know, sorry). As funções básicas, do core do R, serão utilizadas para este propósito. Resumindo, será crucial manipulação de objetos tanto no workspace quando no diretório de trabalho, usando para isso e outros propósitos as funções-base do R, com destaque no uso de loops, condicionais e indexações.

Finalizo respondendo alguns questionamentos pertinentes ao teu comentário a seguir. Sobre esconder a função do processo que pretendo executar: para rodar a função proposta o usuário deverá conhecer as funções do pacote rehh, logo, meio que terá que conhecer o processo sendo executado por trás. A função visa facilitar as análises em com parâmetros estatísticos padrões para os dados, acredito que seria útil para muitas pessoas da área, principalmente após contato inicial com o pacote rehh. Sobre focar na parte da análise ou visualização dos dados: como disse, quem irá analisar mesmo e gerar o output de visualização dos dados serão as funções do pacote rehh (que serão chamados internamente na função que irei criar). Contudo, sobre focar em um tipo de análise em particular, é algo que pode ser feito, e não daria tanto trabalho na correção por parte de vocês caso aceitem esta proposta. Poderia, por exemplo, chamar a função data2haplohh para criar os objetos e utilizar os objetos e lista de SNPs para calcular/plotar o EHH com a função "calc_ehh", mesmo porque os processos que deverão ser realizados para rodar as outras funções serão similares. Sobre as funções do trabalho final deverem ser baseadas nos pacotes básicos: como disse, não constava isso na página de instruções (ainda que se falou sobre usos mais genéricos) e, como pode-se perceber, as funções do pacote rehh seriam utilizadas estritamente para rodar a estatística, que não é o foco da função proposta, não irá tirar quaisquer tipos de trabalhos manuais a serem realizados com diversas funções básicas. Meu intuito ao propor esta função foi unir o útil, que futuramente poderia – sendo otimista– gerar uma publicação após aprimoramentos da função, ao "agradável", que seria os desafios de programação e uso, acredito que intenso, de noções aprendidas ao decorrer do curso e como autodidata, como manipulação de arquivos, objetos, loops, condicionais, indexação e afins. Porém eu sou mero iniciante e aluno da disciplina, cabe a vocês o respaldo se realmente vale a pena ou não prosseguir com esta proposta.

Proposta 2 – Reestruturada

Função: `pop.handling()`

Introdução: Considerando dados provenientes de SNP-arrays, é comum na genética populacional que todos os indivíduos das populações de estudo estejam plotados em um mesmo arquivo com seus genótipos (identificação do SNP e o alelo presente), conforme o cromossomo. Para realizar as análises, quaisquer que sejam, é comum que separemos (extraímos) as populações, gerando novos arquivos para cada uma delas, apenas com os indivíduos respectivos à população de interesse, isso porque nem sempre queremos/precisamos analisar todas as populações. O mesmo se aplica aos SNPs, cujo objetivo nem sempre é analisar todos, e esta mesma função poderia também ser aplicada para extrair SNPs de uma providos por uma lista. Existem atualmente diversos softwares que realizam tais extrações, contudo, sempre para um formato específico, de forma que considerando que o arquivo seja uma matriz (geralmente é), e fornecendo ao usuário a possibilidade de indicar onde está o vetor com as IDs (populacionais ou SNPs), torna-se desnecessário a conversão constante de arquivos por parte destes softwares.

Objetivo: Extrair populações específicas e seus respectivos dados genéticos de um arquivo – matriz – com todas as populações.

Argumentos de entrada:

obrigatórios

- `data` = <matriz com todas as populações e respectivos dados genéticos>
- `location = x.y` <localização do nome dos indíviduos ou ID do SNP no arquivo data, sendo que o primeiro número X (inteiro) pode ser 1 (linha) ou 2 (coluna) e, a fração Y, representa qual linha/coluna que se encontram as IDs; exemplo: se plotado 2.1, então a função irá considerar a primeira (.1) coluna (2) como vetor da matriz que contém as IDs>. *Demais exemplos:*
  1.5 = Vetor está em uma linha; quinta linha.
  2.4 = Vetor está em uma coluna; quarta coluna.
  And so on...
- `x =` <arquivo contendo o nome dos indivíduos da população a ser extraída / ou IDs dos SNPs a serem extraídos independente da população>

opcionais

- `poplot = F` <argumento falso por padrão, mas se verdadeiro, retornaria um gráfico com o N de indivíduos na população (ou SNPs) de interesse comparado com o total>.
- `output = F` <possibilita o usuário definir um nome e gerar diretamente um arquivo .txt que será a saída da função>

Exemplos de arquivos de entrada, como matrizes, são arquivos comumente utilizados na área, como: .vcf .ped .haps

Arquivos/objetos de saída:

- Objeto com a população de interesse e seus dados genéticos
- Gráfico representativo do N de indivíduos da população de interessa comparado com o total do arquivo com todas os indivíduos de todas as populações (opcional)
- Arquivo .txt nomeado pelo usuário com a população de interesse e seus dados genéticos (opcional)

Sumário final: De forma resumida, a função visa comparar os nomes de uma lista – fornecida pelo usuário – com um vetor de uma matriz, cuja localização deste vetor também é indicada pelo usuário, e então "imprimir" todos os dados da matriz que dê match com os nomes da lista, ignorando os dados atrelados aos nomes não encontrados (incluindo eles mesmos). Função aparentemente simples, e por isso estou também super aberto à sugestões para aprimorá-la (quiçá analisar percentagens de NAs ou 0s e afins).

> Eis as propostas modificadas, enviar-te-ei um e-mail também, e qual indicares, mando bala! Gratidão pelas recomendações/críticas (estarei ativo no wiki!).
> — *Cainã Max Couto-Silva 2017/06/12 15:00*

> Oi Cainã, concordo com o que o Gustavo disse e acho que a versão estruturada do plano B é a proposta mais promissora. Além de extrair os dados, acho que seria bom se sua função já calculasse algumas informações iniciais simples sobre cada posição e juntasse tudo numa matriz, coisas como:
>
> - se existe polimorfismo ou não - número de alelos variantes - alguma medida de variabilidade genética.

Danilo G Muniz

Olá Danilo! Acabo de responder o Gustavo via e-mail. Repassando a resposta para ti, irei trabalhar nisso então, e agradeço muito o respaldo de vocês. Qualquer coisa retorno contato convosco. Abraços! — *Cainã Max Couto da Silva 2017/06/12 17:20*

Trabalho Final

Help da função: pop.handling()

```
pop.handling                    package: -                    R Documentation


Extrair indivíduos/SNPs e seus respectivos dados de um arquivo independente
da formatação da estrutura do arquivo, portanto que seja matriz/dataframe.


Description:

    A função requer dois arquivos de entrada - uma matriz/dataframe e uma
lista de identificadores - e a determinação do eixo
    do vetor onde estão os indivíduos/SNPs. A partir da posição do vetor na
matriz definida pelo usuário, compara-se e extrai
    apenas os indivíduos/SNPs presentes na lista de comparação, e todos os
seus dados associados. Argumentos adicionais incluem
    manipulação de NAs, comparação gráfica entre os dataframes
(entrada/saída) e possibilidar de salvar diretamente o novo dataframe.

Usage:

    pop.hanlding(data,axis,list.names, na.omit = F, na.start = 1, graph= F,
out= F)


Arguments:

    data: matriz ou data.frame.
    axis: "c:i" ou "l:i", onde "c" determina que o vetor de nomes para o
match está em coluna no dataframe, "l" indica que está em linha,
    e "i" deve ser um número inteiro indicando a posição ordinal da
coluna/linha onde se encontra o vetor.
    list.names: vetor com os nomes do match de interesse.
```

```
    na.omit: omite NAs se o parâmetro for TRUE (padrão = FALSE), podendo
ainda o usário definir um número (e.g. zero) que será considerado como NA.
    na.start: determina a partir de que linha/coluna os NAs serão
considerados para quantificação/omissão.
    graph: se verdadeiro (TRUE), geram gráficos de comparação entre
```