Comunicado Técnico 219
ISSN 1678-961X
Dezembro, 2014
Santo Antônio de Goiás, GO

Utilização de Ferramentas Computacionais para Análise de Expressão de Genes

Marcelo Gonçalves Narciso[1],
Rosângela Bevitori[2]

Introdução

Nos últimos anos, houve um importante salto na tecnologia de sequenciamento com a introdução de novos conceitos de sequenciamento em larga escala (ELLEGREN, 2008). Os métodos de sequenciamento de "próxima geração" (*next generation sequencing* - NGS) têm a habilidade de processar massivamente sequências em paralelo, sendo capaz de produzir milhões de leituras de sequências de DNA em uma única corrida (CHEN; RATTRAY, 2006). Nesses métodos, as sequências são produzidas diretamente a partir de "bibliotecas" de DNA fragmentado, que não passam pelo processo de clonagem convencional, o que reduz o trabalho, os custos, os erros associados à manipulação de clones, e possibilita a recuperação de transcritos raros ou instáveis quando clonados em bactérias (KALAVACHARLA et al., 2011). Além disso, possibilitam que amostragens de diferentes procedências (tecidos) possam ser analisadas ao mesmo tempo, e posteriormente distinguidas computacionalmente (NOVAES et al., 2008).

Um dos grandes enfoques abrangendo as tecnologias de sequenciamento de nova geração envolve a obtenção não só do genoma, mas também do transcriptoma dos organismos. A obtenção do transcriptoma consiste na identificação do conjunto total de RNAs transcritos de determinado organismo (PASSOS et al., 2000). No âmbito da genômica funcional, o estudo do transcriptoma pode auxiliar na elucidação das funções dos genes. Isto é possível devido ao conhecimento existente sobre expressão gênica de que ela ocorre de maneira diferenciada nos diferentes tecidos, nas diferentes fases fenológicas, em geral, nas diferentes situações em que o organismo se encontra. Como a expressão gênica é dada pela transcrição do DNA, a comparação do perfil de expressão gênica de réplicas biológicas, ou seja, de organismos ou indivíduos completamente idênticos, ou de materiais genéticos contrastantes quanto a uma determinada característica, submetidos a diferentes condições, permitem capturar as diferenças de expressão e os polimorfismos existentes nos genes transcritos, como os *splicings* alternativos e os polimorfismos de único nucleotídeo

[1] Engenheiro elétrico, Ph.D. em Bioinformática, pesquisador da Embrapa Arroz e Feijão, Santo Antônio de Goiás, GO, marcelo.narciso@embrapa.br
[2] Engenheira agrônoma, Ph.D. em Agronomia, pesquisadora da Embrapa Arroz e Feijão, Santo Antônio de Goiás, GO, rosangela.bevitori@embrapa.br

Embrapa

(SNPs), descoberta de genes, perfil, regulação de RNA mensageiro, dentre outras. O sequenciamento de alto desempenho de RNA, também chamado de RNA-seq, é uma das técnicas utilizadas para isso. Esse processo pode permitir não só a compreensão do transcriptoma, mas também a quantificação de todos os genes, a obtenção de suas isoformas (*splicing* alternativo) e a sua completa anotação (GARBER et al., 2011; TRAPNELL et al., 2012).

O estudo do transcriptoma de uma determinada espécie vegetal, associado a esses estresses, encontra uma poderosa ferramenta nas novas tecnologias de sequenciamento que possibilitam varredura massal do genoma a uma velocidade, precisão e escala sem precedentes (MARDIS, 2008), gerando um enorme acúmulo de informações. Estas são analisadas, como um todo, para auxiliar a interpretação biológica dos dados e identificar possíveis genes candidatos associados aos estresses estudados (MAGALHÃES et al., 2010). Torna-se, portanto, essencial uma análise criteriosa dessas informações visando um estudo mais acurado dos genes candidatos a serem identificados.

Entretanto, o grande volume de dados gerados pelo NGS requer as ferramentas computacionais para serem analisados. Nesse sentido, um dos objetivos desta publicação é demonstrar a utilização de dois programas de bioinformática que são amplamente utilizados na análise de transcriptoma: Tophat e Cufflinks (TRAPNELL et al., 2012). Essas são ferramentas livres e de código aberto, produzidas por Trapnell et al. (2012) e utilizadas para análise de genes que permitem a identificação de novos genes, além de novas formas de genes conhecidos, produzidas por *splicing* alternativo, bem como a comparação de genes e de sua expressão diferencial sob duas ou mais condições, através da constatação de diferenças na quantidade de RNA transcrito.

Como exemplo da utilização do Tophat e Cufflinks, foram utilizadas sequências de DNA provenientes do estudo da interação arroz x *Magnaporthe oryzae* (BEVITORI et al., 2010), fungo causador da brusone em arroz, cedidas para este trabalho. Esses dados foram então submetidos à analise e ao final obteve-se quais genes candidados estavam envolvidos no processo de defesa da planta contra o fungo. Isto será mostrado mais adiante neste trabalho.

Como usar esses softwares, em ambiente unix (Linux, Freebsd, AIX, Solaris, HP) e sites de busca e os resultados que são gerados por esses softwares estão descritos a seguir. Para ambiente Windows, esse sistema também poderá ser usado, com as devidas modificações, por ser um ambiente diferente, mas a forma de execução é a mesma que será mostrada para o ambiente unix. Todos os softwares usados são freeware.

Material e Métodos

Nesse estudo, os dados utilizados foram os obtidos por Bevitori et al. (2010) quando sequências de DNA expressas durante a infecção do arroz por *M. oryzae* foram obtidas através do sequenciamento utilizando o equipamento Genome Analyser da plataforma Illumina SBS (Sequencing-by-Synthesis). As sequências foram geradas nos tratamentos 4 e 24 hai (hora após infecção) com um isolado virulento (causa doença) e avirulento (não causa doença) do fungo.

Resultados e Discussão

Nesta secção serão apresentados os softwares usados para as análises e seus resultados, os quais vão mostrar os genes ou *locus* candidatos à resistência ao estresse biótico que a planta foi submetida e será discutido cada passo.

Inicialmente serão vistos os softwares Tophat e Cufflinks para a análise de transcritos e posteriormente ferramentas para análise dos dados gerados, com a finalidade de ver os genes candidatos envolvidos no processo que se deseja conhecer. No caso deste artigo, o processo seria a defesa da planta contra o ataque do fungo virulento.

Tophat é um programa que alinha *reads* (sequências curtas de aminoácidos) a um genoma referência (no caso deste trabalho, a variedade de arroz tropical japônica Nipponbare) com a finalidade de identificar junções exon-exon. Isto é feito graças ao programa para mapear *reads* de comprimento curto (*short reads*) cujo nome é *bowtie* (BOWTIE, 2013).

Tophat pode ser usado com *reads* de qualquer tamanho, com a observação de que, para *reads* com um número menor ou igual a 50 pb, deve-se usar preferencialmente a opção *–bowtie1* (versão 1 do software *bowtie*). Para valores maiores que 50 pb, o mais indicado é usar bowtie2, a versão 2 do software *bowtie*. No caso deste trabalho, como

os *reads* têm 17 pb, será usada a opção *–bowtie1*. Mais sobre o software *bowtie* pode ser visto em Bowtie (2013).

Para usar os *reads* obtidos pelo Genome Analyser, descrito anteriormente, como entrada do software Tophat, basta executar os comandos a seguir.

```
bowtie-build -f ./nipponbare.fasta ./nipponbare
```

```
tophat --bowtie1 -p 16 -r 110 -o ./saidaMetica ./nipponbare ./
metica-f.fastq ./metica-r.fastq
```

O caso acima é para quando se tem os arquivos *fastq forward* (*metica-f.fastq*) e reverso (*metica-r. fastq*). Para o caso *single-end*, com apenas um arquivo *fastq* (no exemplo abaixo, *metica.fastq*) o comando *tophat* é executado da seguinte forma:

```
tophat --bowtie1 -p 16 -r 110 -o ./saidaMetica ./nipponbare ./
metica.fastq
```

Para executar o software Tophat, o primeiro comando é *bowtie-build*, que faz uma indexação do genoma referência (*nipponbare.fasta*), e a saída é o arquivo *nipponbare*, o qual serve de entrada para o comando *tophat*, que é executado logo a seguir. O segundo comando tem a opção *–bowtie1,* que indica a execução do comando *bowtie*, versão 1, para *reads* com menos do que 50 pb. O parâmetro *–p* indica a quantidade de processadores a serem usados, que no caso do exemplo foram 16. A saída produzida pelo comando estará no diretório *saidaMetica*. Os arquivos *metica-f.fastq* e *metica-r.fastq* são fornecidos pelo Genome Analyser, da Solexa/Illumina, já descrito anteriormente. Esses dois arquivos são gerados para o caso do ataque após 4h e também para o caso de 24h. Assim, os comandos acima deverão ser executados duas vezes, uma vez para os dados gerados para 4h e outra vez para os dados gerados para 24h. Após executar o comando *tophat*, no diretório *saidaMetica*, serão gerados os sequintes arquivos com resultados obtidos: *accepted_hits. bam, junctions.bed, insertions.bed* e *deletions. bed.* O arquivo *accepted_hits.bam* é do tipo Binary Alignment/Map (TOPHAT, 2013) e será usado por outros comandos a serem descritos mais adiante. Esse arquivo contém a quantidade de *reads* que foram alinhados com o genoma referência. A forma de cálculo para se conhecer a quantidade de *reads* (relativa ao arquivo de entrada *fastq*) que foram aceitos pode ser vista em (http://vallandingham.me/RNA_seq_differential_expression.html).

Após o comando *tophat* ter sido executado, serão executados os comandos para a montagem dos transcritos (*cufflinks*), mistura de duas ou mais montagens de transcritos (*cuffmerge*) e encontrar genes expressos e os transcritos obtidos de cada ensaio (*cuffdiff*). Os comandos para executar esses passos estão descritos a seguir.

O comando *cufflinks* permite a quantificação dos níveis de expressão de um gene a partir da análise de *reads* com grande acurácia. Essa análise é realizada pelo *assembly* das *reads,* sendo que essa ferramenta é capaz de inferir sobre a estrutura de cada um dos genes e os possíveis *splicings* por ele sofridos utilizando para isso a parcimônia. A análise da expressão é baseada na teoria de que o número de *reads* produzidos também é diretamente proporcional à abundância relativa de transcritos na amostra, normalizando a contagem de transcritos com base no tamanho de cada um deles (TRAPNELL et al., 2012). A maneira de executar o comando *cufflinks* é da seguinte forma, para cada uma das situações (4h e virulento ou 4V, 4h e avirulento ou 4AV, 24h e virulento ou 24V, 24h e avirulento ou 24AV):

```
cufflinks -p 16 -o saidaCuffLinksMetica-4V accepted_hits-4V.bam
cufflinks -p 16 -o saidaCuffLinksMetica-4AV accepted_hits-4AV.bam
cufflinks -p 16 -o saidaCuffLinksMetica-24V accepted_hits-4V.bam
cufflinks -p 16 -o saidaCuffLinksMetica-24AV accepted_hits-4V.bam
```

Os arquivos *accepted_hits-4V.bam*, *accepted_hits-4AV.bam*, *accepted_hits-24V.bam* e *accepted_hits-24AV.bam* são gerados pela saída do comando *tophat*, mencionado anteriormente. Arquivos do tipo *bam* são arquivos do tipo *"binary alignment map"*. Esses arquivos contêm dados diversos dos *reads* lidos pelo comando *tophat* e das montagens de alinhamentos obtidos. O comando *cufflinks* vai gerar, no diretório especificado pela opção "-o" do comando *cuffilinks* o arquivo *transcripts.gtf*, dentre outros. Assim, tem-se então quatro arquivos *transcripts.gtf*, um para cada ensaio.

Esses arquivos serão renomeados para *transcripts-4V.gtf*, *transcripts-4AV.gtf*, *transcripts-24V. gtf*, *transcripts-24AV.gtf*, para não confundir. Um arquivo, contendo os nomes desses quatro arquivos, chamado de *transcritos.txt*, será usado para executar o comando *cuffmerge*, que será descrito mais adiante. Assim, *transcritos.txt* tem o seguinte conteúdo:

```
saidaCuffLinksMetica-4V/transcripts-4Vgtf
saidaCuffLinksMetica-4AV/transcripts-4AVgtf
```

saidaCuffLinksMetica-24V/transcripts-24Vgtf
saidaCuffLinksMetica-24AV/transcripts-24AVgtf

Um detalhe importante sobre o comando *cufflinks* é que este faz uma normalização dos *reads* por transcrito. Pode ser que um transcrito tenha muito mais *reads* que outro transcrito, ou a cobertura deste seja maior que de outro, e assim faz parecer que o transcrito que tem mais *reads* foi o mais expresso. Segundo Trapnell et al. (2012), o número de *reads* gerado a partir de uma transcrição é diretamente proporcional à abundância relativa do transcrito na amostra. No entanto, devido ao fato dos fragmentos de cDNA serem geralmente de tamanho fixo como parte da construção da biblioteca (para otimizar a saída do sequenciador), transcrições maiores produzem mais fragmentos de sequências do que transcritos mais curtos.

Para calcular o nível de expressão correto de cada transcrito, o programa Cufflinks deve contar os *reads* que pertencem a cada transcrito e depois normalizar essa contagem pelo comprimento de cada transcrito. Assim, para comparar o nível de expressão de um transcrito, as contagens devem ser normalizadas e isto é feito também durante a execução do comando *cufflinks*. A fórmula de cálculo pode ser dada pelo cálculo de RPKM (*Reads Per Kilobase per Million mapped reads*), que é dado por:

RPKM = C/(N.L)

Em que, C é o número de *reads* que são mapeados em uma dada característica (transcritos, exon, etc.), N é o número total de *reads* mapeados (em milhões) e L é o número de pares de bases do exon considerado (em kb). A Figura 1 ilustra o cálculo.

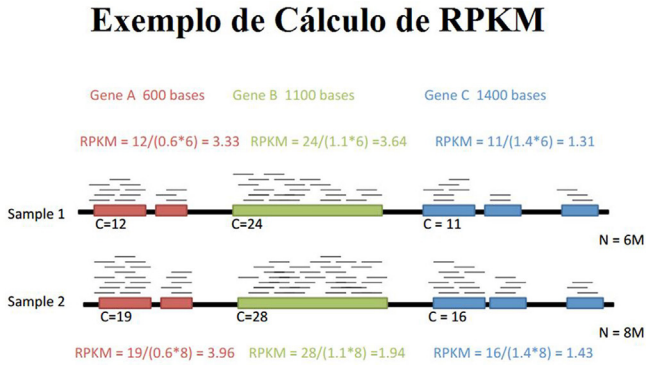

Figura 1. Fórmula para o cálculo de RPKM.

O valor de RPKM pode ser usado como uma medida da abundância de transcritos. Outra forma de medir essa quantidade de transcritos, de forma normalizada, é dada por FPKM, que significa ***F**ragments **P**er **K**ilobase of exon per **M**illion fragments mapped* (**FPKM**). Mais informações podem ser vistas em Trapnell et al. (2012) e Thiru (2013).

Feitas estas considerações sobre o programa Cufflinks, considere o programa Cuffmerge. Para análise de expressão gênica diferencial, quando é utilizada mais de uma condição e obtidos transcritos dessas diferentes condições, provindos de réplicas biológicas, o protocolo de Trapnell et al. (2012) descreve o programa Cuffmerge como sendo utilizado para "fundir" os resultados dos *assemblies* realizados para cada um dos tratamentos descritos (4V, 4AV, 24V, 24AV), visto que se recomenda que o *assembly* de cada amostra seja realizado individualmente. Esse programa é executado da seguinte forma:

cuffmerge -g ./nipponbare.gtf -s ./nipponbare.fasta -p 16 ./ transcritos.txt

O comando acima tem a opção *–g nipponbare.gtf*. Esse arquivo pode ser baixado a partir do site: www.phytozome.net.

O comando *cuffmerge* serve para gerar uma anotação única dos transcriptomas envolvidos. O resultado é um arquivo de nome *merged.gtf*. Após esse comando, é executado o comando *cuffdiff*. O propósito dessa ferramenta é calcular a expressão em duas ou mais amostras e testar a significância das modificações observadas na expressão entre elas, de maneira a verificar se as diferenças observadas, após retiradas do modelo as possíveis fontes de viés, foram devidas às diferentes condições impostas às réplicas biológicas, permitindo a anotação dos genes. A execução do comando está a seguir:

cuffdiff -o diff_out_4 -b ./nipponbare.fasta -p 16 -L C1,C2 -u merged.gtf accepted_hits-4V.bam accepted_hits-4AV.bam

cuffdiff -o diff_out_24 -b ./nipponbare.fasta -p 16 -L C1,C2 -u merged.gtf accepted_hits-24V.bam accepted_hits-24AV.bam

No primeiro comando, são considerados os dados de 4h para o fungo virulento e 4h para o fungo avirulento. O segundo comando considera os dados de 24h para o fungo virulento e 24h para o fungo avirulento. Os parâmetros usados nos comandos acima podem ser vistos em Trapnell et al. (2012). A saída estará nos diretórios *o diff_out_4* e *o diff_*

out_24. Nesses diretórios há vários arquivos contendo estatísticas diversas, e dentre eles o arquivo *gene_exp.diff*. Um trecho desse arquivo está a seguir.

sam-ple_1	sam-ple_2	status	value_1	value_2	log2 (fold)	test_stat	p_value	q_value	Significant
C1	C2	OK	73678.6	10134.7	-286.195	392.247	8,76E-03	0.040614	Yes
C1	C2	OK	1440.66	289.097	-23.171	402.622	5,67E-03	0.0350514	Yes
C1	C2	OK	124.081	993.336	0.320928	0.403514	0.068657	0.955297	No
C1	C2	OK	78292.9	110140	0.492385	0.818113	0.013293	0.936029	No

Com esse arquivo e outros gerados por esse processo, é possível saber quais genes participam do processo de combate ao fungo da brusone e quais não participam ou são silenciados. Esses arquivos são para os ensaios 4V x 4AV e 24V x 24AV.

Falta ainda verificar em que gene a sequência está. Assim, primeiro faz-se um filtro do arquivo *gene_exp.diff* para quando *significant = yes* ou quando *p-value* é menor que certo valor (0.01, por exemplo) e o valor absoluto de *log2(fold)* for maior que um certo valor (2, por exemplo). Assim, usando o arquivo exemplificado, anteriormente, tem-se as regiões do cromossomo que são ativadas:

sam-ple_1	sam-ple_2	status	value_1	value_2	log2 (fold)	test_stat	p_value	q_value	Significant
C1	C2	OK	73678.6	10134.7	-286.195	392.247	8,76E-03	0.040614	Yes
C1	C2	OK	1440.66	289.097	-23.171	402.622	5,67E-03	0.0350514	Yes

C1 e C2 correspondem aos ensaios para 4V e 4AV, para o caso do primeiro comando *cuffdiff* apresentado, e 24V e 24AV para o segundo comando, ilustrado anteriormente. O trecho no qual está descrito onde a sequência está, e que complementa o exemplo dado, é o que está abaixo.

test_id	gene_id	gene	locus	sample_1	sample_2
XLOC_000221	XLOC_000221	-	Chr1:6281506-6281740	C1	C2
XLOC_000891	XLOC_000891	-	Chr1:28731490-28732206	C1	C2
XLOC_000966	XLOC_000966	-	Chr1:30184249-30184382	C1	C2

Para obter as regiões nas quais as sequências obtidas foram escolhidas, é feito o seguinte procedimento, dentre outros possíveis.

1 – obter os cromossomos a partir da variedade referência *nipponbare.fasta* e gerar os arquivos *cromo1.fasta*, ..., *cromo12.fasta*.

2 – criar uma base de dados para busca usando o comando *makeblastdb* para cada um dos doze cromossomos. Por exemplo, para o cromossomo 1 seria o seguinte:

```
makeblastdb -in ./cromo1.fasta  -input_type fasta -dbtype 'nucl' -out  ./cromo1.db
```

3 – Obter as sequências desejadas, a partir do arquivo gerado por *cuffdiff*. Para exemplificar, para o caso do *loci* ser Chr1:6281506-6281740, então é executado:

```
blastdbcmd -db cromo1 -range 6281506-6281740 -dbtype nucl -entry all
```

Esses comandos, *makeblastdb* e *blastdbcmd,* pertencem a um kit de busca do software BLAST feito pelo NCBI (http://www.ncbi.nlm.nih.gov/). A saída do comando *blastdbcmd,* para cada ChrXX:YYY-ZZZ, é uma sequência, tal como o exemplo abaixo:

```
>gnl|BL_ORD_ID|0:26174154-26174382 Chr7

TACGTGGTGCTCGGAGACGGCGGCCGCGGCGGGGAGAGAGG
GGAGCCATGGAACATTAGGATAGAGGACGTGGATTAGGACC
AGGATTATATTCTTCAAAT
```

Com essa sequência, faz-se então uma busca através do gene ao qual a sequência pertence usando o comando *blastn*, tal como está feito abaixo, considerando-se que essa sequência está em um arquivo chamado *saidaCuff.fasta*. O comando para isto é:

```
blastn -db /usr/local/db/nt -query  saidaCuff.fasta -num_threads 16
```

A respeito dos comandos do blast (blastn, blastx, update_blastdb.pl, etc.), estes podem ser baixados a partir do site ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/.

Sobre a base de dados nt, esta pode ser baixada a partir de ftp://ftp.ncbi.nlm.nih.gov/blast/db/. Para o exemplo dado, a base de dados nt está disponível em /usr/local/db.

Com o uso dos comandos do blast, o processo se dá de maneira mais rápida, em relação ao processo feito a partir da página http://blast.ncbi.nlm.nih.gov/Blast.cgi, pois todas as sequências podem ser alinhadas de uma vez.

Para o comando acima, a opção –num_threads 8 significa ter 16 threads, isto é, dividir o processamento em dezesseis processos.

Outra forma de busca do gene pode ser feita através do site do NCBI (http://blast.ncbi.nlm.nih.gov/Blast.cgi) usando a sequência desejada.

Após a busca, são obtidos os genes, que têm a identificação do tipo, LOC_Os0RgXYZKW, R,X,Y,Z,k,W e são números de 0 a 9.

Desde o começo da execução com o *tophat* até o comando *cuffdiff*, pode ser feito um *script*, isto é, um arquivo que contém todos os comandos para execução. Seja o *script* com o nome *genes.sh*. O conteúdo desse *script*, que são os comandos descritos até aqui, é o seguinte:

```
#!/bin/bash
bowtie-build -f ./nipponbare.fasta ./nipponbare
tophat --bowtie1 -p 16 -r 110 -o ./saidaMetica ./nipponbare ./metica-f.fastq ./metica-r.fastq
cufflinks -p 16 -o saidaCuffLinksMetica-4V accepted_hits-4V.bam
cufflinks -p 16 -o saidaCuffLinksMetica-4AV accepted_hits-4AV.bam
cufflinks -p 16 -o saidaCuffLinksMetica-24V accepted_hits-4V.bam
cufflinks -p 16 -o saidaCuffLinksMetica-24AV accepted_hits-4V.bam
echo "saidaCuffLinksMetica-4V/transcripts-4Vgtf" > ./transcritos.txt
echo "saidaCuffLinksMetica-4AV/transcripts-4AVgtf >> ./transcritos.txt
echo "saidaCuffLinksMetica-24V/transcripts-24Vgtf >> ./transcritos.txt
echo "saidaCuffLinksMetica-24AV/transcripts-24AVgtf >> ./transcritos.txt
cuffdiff -o diff_out_4 -b ./nipponbare.fasta -p 16 -L C1,C2 -u merged.gtf accepted_hits-4V.bam accepted_hits-4AV.bam
cuffdiff -o diff_out_24 -b ./nipponbare.fasta -p 16 -L C1,C2 -u
merged.gtf accepted_hits-24V.bam accepted_hits-24AV.bam
```

Após fazer as considerações sobre o arquivo *gene_exp.diff*, e gerar as bases cromo1db até cromo12.db, usando *makeblastdb*, um outro *script* pode ser utilizado para gerar os genes com as ferramentas do *blast*. Supondo que, após algum critério de análise, o usuário deixe nesse arquivo a coluna *"significant"* com os campos desejados iguais a *"yes"*, então, o *script* para determinar os genes candidatos seria:

```
#!/bin/bash
grep -i "yes" gene_exp.diff > gene_exp_filtered.diff
    while read line; do
      cromo="Chr"
      export cromo
      existeCromo=`echo -e "$line" | awk '{ if( index($0, ENVIRON["cromo"]) > 0 ) {print "1"} else {print "0"} }' `
      if [ "$existeCromo" -eq 1 ]
          then
            doisPontos=`echo -e "$line" | awk '{ if ( substr($0, index($0, $cromo) + 4, 1) == ":") {print "1"} else {print "0"} }' `
            if [ "$doisPontos" -eq 1 ]
                then
                  numCromo=`echo -e "$line" | awk '{ print substr($0, index($0, "Chr") + 3, 1) }' `
                else
                  numCromo=`echo -e "$line" | awk '{ print substr($0, index($0, "Chr") + 3, 2) }' `

            fi # fim de if [ "$doisPontos" -eq 1 ]
            aux=`echo -e "$line" | awk '{ print $4}' `
            posInic=`echo -e "$aux" | awk '{ print index($0,":") + 1}' `
            posFinal=`echo -e "$aux" | awk '{ print index($0,"-") - 1}' `
            export aux
            export posInic
            export posFinal
            limInf=`echo -e "$aux" | awk '{ print substr( ENVIRON["aux"], ENVIRON["posInic"], ENVIRON["posFinal"] - ENVIRON["posInic"] + 1 ) }' `
            limSup=`echo -e "$aux" | awk '{print substr($0, index($0,"-") + 1)}' `

        fi # fim de if [ "$existeCromo" -eq 1 ]

      base=`echo "cromo"$numCromo`
      range=`echo $limInf"-"$limSup`
      blastdbcmd -db $base -range $range -dbtype nucl -entry all > sai.fasta
      saidaBlast=`echo "cromo"$numCromo"-"$range".txt" `
        blastn -db nt -query sai.fasta -num_threads 16 > $saidaBlast
    done < gene_exp_filtered.diff # final do commando while read line; do
rm gene_exp_filtered.diff
```

A Figura 2 ilustra o processo envolvendo *tophat* e *cufflinks*, conforme Trapnell et al. (2012).

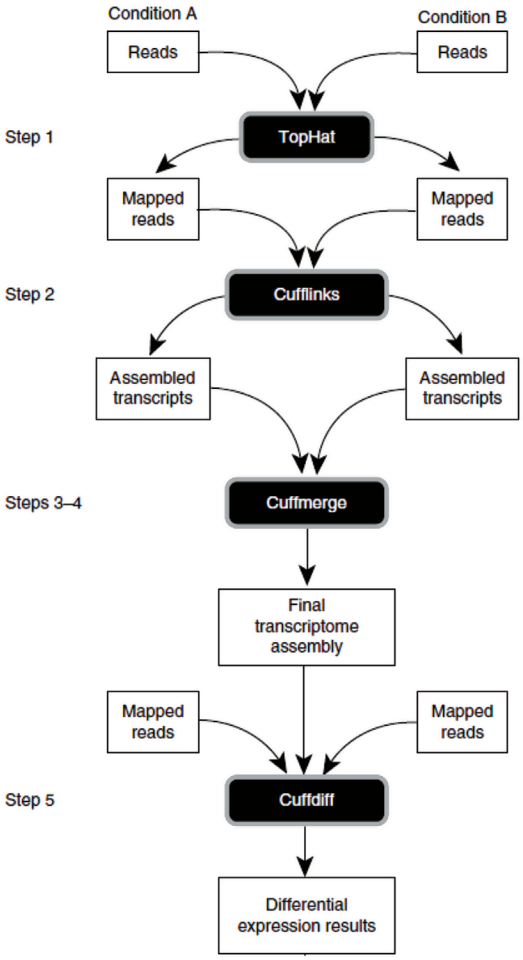

Figura 2. Fluxograma do processo de extrair resultados a partir dos *reads* em arquivos *fastq.*

Após ser executado esse *script*, existirão os genes ou *locus* candidatos em vários arquivos com nomes do tipo cromoXX-YY-ZZ.txt. XX é o número do cromossomo, YY é o valor do intervalo inferior e ZZ é o valor do intervalo superior. Esses arquivos foram gerados em um trecho do *script* com os comandos:

```
saidaBlast = `echo "cromo"$numCromo"-"$range".txt" `

blastn -db /usr/local/db/nt -query  sai.fasta -num_
threads 16 >  $saidaBlast
```

Em cada um desses arquivos tem o loci ou gene associado ao cromossomo XX, com intervalo inferior YY e superior ZZ.

Com esses genes, faz-se uma busca sobre a função de cada um usando o site http://bioinfo.cau.edu.cn/agriGO. Nesse site, para essa análise, é usada a opção *Analysis Tool*. Ao acessar essa opção, basta passar um conjunto de genes que o site mostra em que atividade cada gene participa, conforme ilustra a Figura 3.

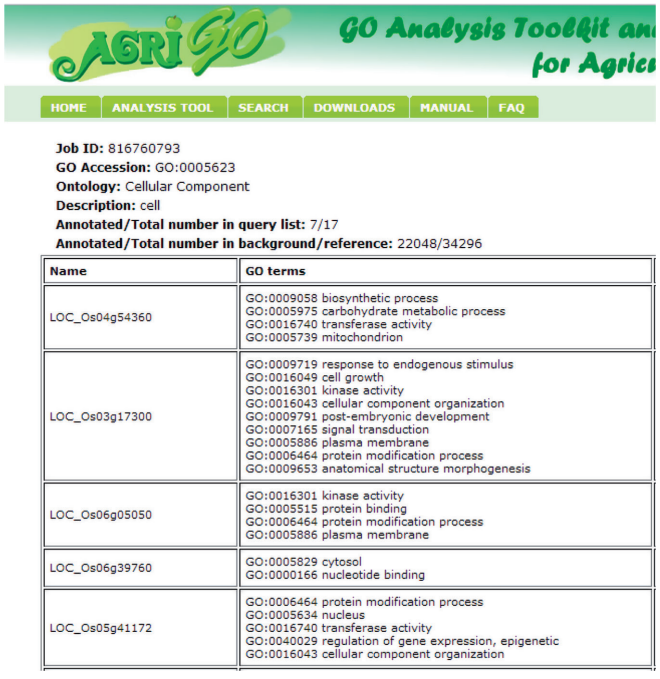

Figura 3. Resultados de genes usando o site agriGO.

São vários genes envolvidos, e todos aqueles que tiverem GO *terms "stress response"* ou *"response to biotic stimulus"* ou *"biotic stress"* também são genes que podem ser considerados candidatos a serem validados.

Outra forma para determinar a função de cada gene pode ser feita através do software blast2go, que pode ser acessado em: www.blast2go.org.

Conclusão

Este trabalho mostrou uma forma de uso de ferramentas computacionais para encontrar genes envolvidos em resposta ao estresse. Para o caso deste trabalho, foi enfocado o fungo que causa a brusone. Os softwares usados para a análise dos dados são gratuitos e qualquer pessoa poderá realizar esse procedimento para uma situação similar. O tempo de execução desses programas depende da quantidade de processadores do computador que vai rodar os programas. O tempo médio para executar essas tarefas de análise por *tophat* e *cuffdiff* é de um a três dias. Se o computador tiver apenas um processador, o tempo vai ser maior, mas a análise será feita da mesma forma.

Referências

BEVITORI, R.; MEYERS, B. C.; PAPPAS JUNIOR, G. J.; NAKANO, M.; JAIN, G.; FRAGOSO, R. R.; LOURENCO, I. T.; GROSSI-DE-SA, M. F. Application of next generation sequencing to study rice blast disease. In: INTERNATIONAL RICE BLAST CONFERENCE, 5., 2010, Little Rock, Arkansas. **Proceedings**. Little Rock: USDA-ARS: University of Arkansas, 2010. p. 69.

BOWTIE. Disponível em <http://bowtie-bio.sourceforge.net/index.shtml>. Acesso em: 25 set. 2013.

CHEN, J.; RATTRAY, M. Analysis of tag-position bias in MPSS technology. **BMC Genomics**, v. 7, n. 77, Apr. 2006.

ELLEGREN, H. Sequencing goes 454 and takes large-scale genomics into the wild. **Molecular Ecology**, Oxford, v. 17, n. 7, p. 1629-1631, Apr. 2008.

GARBER, M.; GRABHERR, M. G.; GUTTMAN, M.; TRAPNELL, C. Computational methods for transcriptome annotation and quantification using RNA-seq. **Nature Methods**, New York, v. 8, n. 6, p. 469-477, June 2011.

KALAVACHARLA, M.; LIU, Z.; MEYERS, B. C.; THIMMAPURAM, J.; MELMAIEE, K. Identification and analysis of common bean (*Phaseolus vulgaris* L.) transcriptomes by massively parallel pyrosequencing. **BMC Plant Biology**, v. 11, n. 135, Oct. 2011.

MAGALHÃES. J. P. de; FINCH, C. E.; JANSSENS, G. Next-generation sequencing in aging research: emerging applications, problems, pitfalls and possible solutions. **Ageing Research Reviews**, Kidlinton, v. 9, n. 3, p. 315–323, July 2010.

MARDIS, E. R. The impact of next-generation sequencing technology on genetics. **Trends in Genetics**, Amsterdam, v. 24, n. 3, p. 133-141, Mar. 2008.

NOVAES, E.; DROST, D. R.; FARMERIE, W. G.; PAPPAS, G. J.; GRATTAPAGLIA, D.; SEDEROFF, R. R.; KIRST, M. High-throughput gene and SNP discovery in *Eucalyptus grandis*, an uncharacterized genome. **BMC Genomics**, v. 9, n. 312, June 2008.

PASSOS, G. A. S.; NGUYEN, C.; JORDAN, B. Projeto Transcriptoma: análise da expressão gênica em larga escala usando DNA-arrays. **Biotecnologia Ciência & Desenvolvimento**, Brasília, DF, v. 2, n. 12, p. 34-37, jan./fev. 2000.

THIRU, P. **RNA-Seq**: methods and applications. Disponível em: <http://jura.wi.mit.edu/bio/education/hot_topics/RNAseq/RNA_Seq.pdf>. Acesso em: 22 nov. 2013.

TOPHAT. Disponível em: <http://tophat.cbcb.umd.edu/manual.shtml#toph>. Acesso em: 25 set. 2013.

TRAPNELL, C.; ROBERTS, A.; GOFF , L.; PERTEA, G.; KIM, D.; KELLEY, D. R.; PIMENTEL, H.; SALZBERG, S. L.; RINN, J. L.; PACHTER, L. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. **Nature Protocols**, v. 7, n. 3, p. 562-578, Mar. 2012.

Comunicado Técnico, 219

Exemplares desta edição podem ser adquiridos na:
Embrapa Arroz e Feijão
Endereço: Rod. GO 462 Km 12 Zona Rural, Caixa Postal 179 75375-000 Santo Antônio de Goiás, GO
Fone: (62) 3533 2123
Fax: (62) 3533 2100
www.embrapa.br/fale-conosco/sac
1ª edição
Versão online (2014)

Comitê de publicações

Presidente: *Pedro Marques da Silveira*
Secretário-Executivo: *Luiz Roberto R. da Silva*
Membros: *Camilla Souza de Oliveira, Luciene Fróes Camarano de Oliveira, Flávia Rabelo Barbosa Moreira, Ana Lúcia Delalibera de Faria, Heloisa Célis Breseghello, Márcia Gonzaga de Castro Oliveira, Fábio Fernandes Nolêto*

Expediente

Supervisão editorial: *Luiz Roberto R. da Silva*
Revisão de texto: *Camilla Souza de Oliveira*
Normalização bibliográfica: *Ana Lúcia D. de Faria*
Editoração eletrônica: *Fabiano Severino*

CGPE: 11469