Uma nova proposta de busca por tandem repeats

Narciso, M.G(1); Yamagishi, M.E.B.(2)
narciso@cnpaf.embrapa.br
(1)Biotechnology Department, Embrapa Arroz e Feijão, Goiania-GO; (2)Bioinformatic Department, Embrapa Informática Agropecuária, Campinas-SP.

RESUMO

Tandem Repeats (TR) são sequências onde os mesmos padrões se repetem consecutivamente e têm sido usado como marcadores genômicos (microssatelite e minissatelite) há muitos anos. Recentemente, novos estudos têm associado TR a importantes processos regulatórios o que aumentou consideravelmente o interesse em TR. A redução exponencial do custo de sequenciamento causado por novas tecnologias resultou em proliferação de projetos relacionados a genoma e assim a busca por marcadores moleculares, tais como microssatélites e minissatélites (TR) e SNPs, se torna cada vez maior. Assim, é importante ter um algoritmo mais rápido para busca de TR em genoma e também que forneça o maior número possível de TR. Neste trabalho é proposta uma nova estratégia para se obter TR de forma mais rápida e também com eficiência. Esta nova proposta, que sera chamada de ConvolutionTR, reduz o tempo de busca por TR em torno de 30% a 50% em relação a outros enfoques conhecidos da leiteratura e também consegue obter uma quantidade maior de TR.
PALAVRAS-CHAVE: tandem repeats, algoritmo, desempenho.

INTRODUÇÃO

Com a conclusão de diversos projetos relacionados a genomas, vieram à tona alguns fatos inesperados como, por exemplo, a constatação que, aproximadamente, 50% dos genomas de mamíferos são compostos por seqüências repetitivas (Bannert & Kurth, 2004). Nas plantas essa proporção pode alcançar fenomenais 90% (Kazazian, 2004; SanMiguel et al., 1996). As seqüências repetitivas podem ser classificadas como tandem repeats (TR) ou interpersed repeats. São exemplos do primeiro grupo os microsatélites, minisatélites e telômeros; do segundo grupo, os elementos genéticos móveis ou, simplesmente, transposons. Há aplicações instrumentais para os dois grupos. Os minisatélites são utilizados como marcadores moleculares em estudos de melhoramento genético de plantas e de animais para identificação de genes de interesse científico ou econômico (Lander & Botstein, 1989; Sharma et al., 2007), ou, mais comumente, em testes de paternidade (Botstein et al., 1980;Rocheta et al., 2007); já os transposons têm aplicações na regulação gênica e terapia genética (Ivics & Izsvak, 2006).

Identificar as seqüências repetitivas em um genoma pode ser visto como um problema de alinhamento múltiplo de seqüências, que é um problema NP-Hard (Wang & Jiang, 1994; Jiang et al., 2000). O simples alinhamento local de duas seqüências já é um problema computacionalmente caro (Waterman & Smith, 1981), e exigiu o desenvolvimento de heurísticas como o

FASTA (Wilbur & Lipman, 1983) e o BLAST (Altschul et al., 1990) para acelerar, e, conseqüentemente, tornar viável a busca em bases de dados gigantescas.

Há uma coleção significativa de softwares para identificar seqüências repetitivas. Talvez o mais popular seja o Tandem Repeat Finder (http://tandem.bu.edu/trf/trf.html). Entretanto, dada a complexidade do problema de identificar as seqüências repetitivas, a quantidade de novos softwares e metodologias cresce a cada ano (Zhi et al., 2006; Price et al., 2005; Kurtz et al., 2001). Uma observação a ser feita é que cada um destes softwares não consegue obter todas as seqüências possíveis ou ainda são limitados a um tamanho máximo de seqüência (não conseguem rodar para um genoma de arroz, por exemplo) ou necessitam de grande quantidade de processamento e memória para serem executados. Enfim, existem limitações nos algoritmos e assim ainda são necessárias pesquisas com a finalidade de se obter algoritmos que possam ser executados de forma a obter todos os TR de uma seqüência, por maior que ela seja, e também em tempo razoável.

Este trabalho mostra um algoritmo, de propósito geral, para se obter todas as TR, qualquer que seja o tamanho, em uma seqüência qualquer. Uma das vantagens deste algoritmo é que não necessita de estrutura de dados complexas e também não necessita de grande quantidade de memória e é executado rapidamente, conforme poderá ser visto nos resultados a serem mostrados neste trabalho.

MATERIAIS E MÉTODOS

A ideia central deste algoritmo é como uma seqüência é comparada com ela mesma, defasada de k posições. Assim, para o caso do DNA, se k for igual a 1, será possível obter as repetições de uma mesma base nitrogenada A, T, C ou G (período da seqüência é igual a 1). Se k for igual a 2, será possível obter a repetição de sequencias de duas bases (por exemplo, AT, CT, GC, AG, etc.). Se k for igual a 3, será possível obter uma seqüência de repetições de 3 bases (pelo menos uma destas é diferente e o período é 3), e assim por diante, valendo para k = n. Para exemplificar, seja a seqüência abaixo.

TCGGATGATTTTTTATAG

Comparando a mesma seqüência com o atraso de uma posição, tem-se o seguinte:

```
T C G A T G A T T T T T T A G
  T C G A T G A T T T T T T A
0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0
```

A primeira e segunda linhas são as mesmas com um atraso de uma base da primeira em relação a segunda linha. A terceira linha é composta somente por 0 e 1, e 0 indica comparação de bases diferentes e 1 indica comparação de bases iguais. Observe que existe uma seqüência de TTTTTT e a terceira linha indica 111110. Por causa do atraso das duas seqüências, o número de valores iguais a 1 é uma unidade a menos que a quantidade da repetição. Neste caso, para 6 repetições (TTTTTT), tem-se 5 repetições de 1 (11111). De forma análoga, seja a seqüência abaixo, para exemplificar a repetição de uma seqüência repetida de duas bases.

GATGATATATATATATC

Se compararmos esta seqüência com ela mesma com o atraso de duas bases, tem-se a seguinte comparação.

```
G A T G A T A T A T A T A T C
    G A T G A T A T A T A T A T A
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0
```

Observe que novamente, temos uma seqüência (ATATATATATAT), com período de duas bases e frequência de 6 repetições. Observe que se teve 10 repetições de 1 (1111111111) na terceira linha, que mostra a comparação entre as duas seqüências com atraso de duas bases. Como o atraso é de duas bases, e tem-se 10 repetições, então basta dividir 10 por 2 (número do atraso, que indica o período da seqüência desejada). Assim, tem-se 5, que é a freqüência da seqüência menos 1 (o atraso).

Assim, de forma geral, se tiver uma seqüência de período p, e frequência de repetições f, e k for o número do atraso, tem-se que, se o número n de comparações iguais a 1 obtidos for igual a $n=(f*k - k) = (f - 1)*k$ e $k = p$, então temos uma seqüência periódica, com período p (igual ao atraso k) e frequência f. Matematicamente, seria

$$f = (n/k) + 1 \quad e \quad p = k$$

O valor da divisão de n/k deve ser transformado para inteiro, desprezando-se a casa decimal.

Com esta ideia central, é possível de se fazer um algoritmo rápido para se obter as seqüências de repetições desejadas. Pode-se fazer uma laço de repetição (loop) com i variando de 1 a S (período máximo de busca) e comparar a seqüência com ela mesma com o atraso i. De forma geral, seria seqüência (j) versus seqüência (j – i) (uma comparação), j variando de i a F, F é a posição final da seqüência (tamanho da seqüência). Assim, para cada valor de i, ter-se-ia seqüências de 1 ou 0, resultado da comparação, e cada uma das seqüências de 1 seria avaliada para ver se a divisão da freqüência desta seqüência por i seja exata ou a parte inteira maior que um valor pré-estabelecido. Observe que se procura a seqüência de período mínimo i (múltiplos de i não valem).

Um detalhe a ser levado em conta é que se podem ter períodos nos quais existem outros períodos. Por exemplo, seja a seqüência abaixo.

AAAAATAAAAATAAAAATAAAA
ATAAAAAT

A seqüência tem um período AAAAAT. Porém, este período tem o período AAAAA. Assim, quando no atraso de uma base (k = 1), o algoritmo vai encontrar a seqüência AAAAA. Quando o atraso for igual a 6 (k = 6), o algoritmo vai encontrar a seqüência AAAAAT. O algoritmo mostra todos esses períodos e frequências desde que sejam mínimos. Assim, mostraria os períodos A e AAAAAT, com a respectiva posição de início. Fica a cargo de quem vai analisar ver qual o período mínimo que começa na dada posição que vai ser considerado.

Sobre período mínimo, para melhor entendimento, seja a seqüência CAAAAAAAAAAT. Essa sequencia poderia ter os períodos A (frequência 10), AA (frequência 5), AAA (frequência 3), porém, como o período mínimo é 1, então esse é o valor que será mostrado pelo algoritmo. Se uma seqüência é múltipla de uma que foi já obtida, então esta seqüência que se repete será descartada em favor da que tiver um período mínimo (claro que se referem as mesmas bases da seqüência em estudo).

O algoritmo para este tipo de análise de seqüência para obter tandem repeats seria.

Algoritmo ConvolutionTR

Ler a sequencia de interesse. Seja F o tamanho da seqüência.

TamanhoMinimoDaFrequencia = 5; // altere aqui se quiser uma frequência maior

Repita para i = 1, 2, ..., S, S é o período máximo desejado

contador_de_1 = 0;

Repita para j = i, i+1,i+2, ...F - i

Se (sequencia(j) = sequencia(j – i))

então contador_de_1 = contador_de_1 + 1;

senão

contador_de_1_anterior = contador_de_1;

contador_de_1 = 0

Fim_Se

Se (contador_de_1 = 0 e (i+contador_de_1_anterior / i) >= TamanhoMinimoDaFrequencia)

então

pos = j - i* contador_de_1 + i;

//

// Imprimir resultado parcial

imprimir ("posicao inicial >>" + pos)

repita para r = pos, pos +1,...j-1

imprima (seqüência(r))

fim_repita

Fim_Se

Fim_Repita para j

Fim_Repita para i

Acima está o algoritmo, de forma simplificada, do que seria para se obter tandem repeats de uma seqüência. Obviamente, outros detalhes, como, por exemplo, verificar se a seqüência obtida é múltipla de uma outra seqüência já obtida anteriormente, não estão descritos acima por amor a simplicidade.

O que geralmente se tem são seqüências grandes, cujo tamanho podem ser de dezenas de bytes a alguns GB. Assim, não é possível ler uma seqüência muito grande em uma só variável (infelizmente) e executar o algoritmo acima. O que é feito é ler a seqüência por partes, do início ao fim, e para cada parte lida executar o algoritmo acima. Para se fazer isso, tem que se considerar a seqüência anterior lida e a posterior a ser lida pois uma seqüência pode ter começado no final da seqüência anteriormente lida ou ainda, não terminar na seqüência atual. Enfim, tem mais casos a serem considerados. Assim, imagine um arquivo que tenha uma seqüência com 1.000.000 de bases. Admita que esse tamanho seja dividido 100 leituras de 10.000 bases. Imagine, sem perda de generalidade, que se deseja obter seqüências de até 100 bases como período e que se esteja na leitura do décimo pedaço da seqüência total. Assim, tem-se que se levar em conta se a seqüência anterior, ao ser analisada, terminou em algum atraso k com o valor 1. Se terminou, isso significa que a seqüência de 1 ainda pode continuar na seqüência atual que está sendo analisada. Então, todas as seqüências parciais de 1 que vieram da seqüência anterior deverão ser guardadas (claro que aquelas que se referem ao final da seqüência anterior e que vão continuar na seqüência atual). Esse detalhe tem que ser adicionado ao algoritmo acima, pois ele é para o caso de quando a seqüência é pequena suficiente para ser guardada em uma variável. O desempenho do algoritmo será tanto melhor quanto maior for a parcela da seqüência total que puder ser

armazenada na variável (quanto mais memória, melhor).

RESULTADOS E DISCUSSÃO

O algoritmo da proposta deste trabalho foi feito usando a linguagem C e pode ser executado em qualquer plataforma, após ser compilado, visto que só depende de bibliotecas de C que existem em todos os compiladores. Este programa então foi comparado com alguns sistemas para se obter tandem repeats, que podem ser vistos em http://en.wikipedia.org/wiki/Tandem_repeat. Estes sistemas são TRF (Tandem Repeat Finder), disponível em http://tandem.bu.edu/trf/trf.html, MREPS, disponível em http://bioinfo.lifl.fr/mreps/, SWAN (Structure Word Analyser), disponível em http://favorov.imb.ac.ru/swan/, PHOBUS, disponível em http://www.ruhr-uni-bochum.de/spezzoo/cm/cm_phobos.htm
O ambiente computacional no qual foram feitos os testes (comparação) é CentOs linux, v 5.5, kernel 2.6.18-194.3.1.el5, 64 bits, 16 Intel(R) Xeon(R) CPUs, 2.80GHz cada CPU, cache igual a 8192 KB.
As Tabelas 1, 2 mostram comparações entre a nova proposta e o algoritmo TRF pata tempo de execução e também para TR obtidos quanto à sua freqüência.

Tabela 1 - TRF x ConvolutionTR – tempo de execução (s)

Sequences	**TRF (s)**	**ConvolutionTR (s)**
Seq 1	1067	570
Seq 2	107	66
Seq 3	300	187
Seq 4	61	41

Na tabela acima, as sequências são:

Seq 1 – DNA da variedade Nipponbare (arroz)
Seq 2 – cromossomo 1 da variedade Nipponbare –
Seq 3 – DNA de Arabdopsis thaliana (At)
Seq 4 - Cromos. 5 de Arab thaliana

Tabela 2 – Frequência de TR x número de sequências repetidas em relação ao DNA da variedade Nipponbare (arroz) e período de TR maior ou igual a 12

Freq TR	TRF	ConvolutionTR	Phobus
4	60312	238004	233819
5	10912	57941	57117
6	3624	20359	19825
7	1661	9924	9510
8	941	5154	4932
9	626	3179	3046
10	424	1997	1941
11	319	1310	1281
12	223	960	944
13	167	744	730
14	140	631	616
15	146	579	579
16	118	467	458
17	103	414	406

Tabela 3 - Comparação de tempo de execução e número de TR obtidos por algoritmos (DNA de Nipponbare)

Software	**Tempo(s)**	**Número de TR**
ConvolutionTR	548	149014
Mreps	30	18460
Phobos	1013	146591
TRF	964	82663

Tabela 4 – Tempo de execução para se obter todos os TR de periodo entre 2 e 100 e frequência acima 8 (DNA Nipponbare)

Swan	13912	2706
Software	**Tempo(s)**	**Número de TR**
Phobos	985	15347
Swan	13912	2987
ConvolutionTR	542	15691
TRF	964	3781
Mreps	19	3448

Tabela 5 – Tempo de execução para se obter todos os TR de periodo entre 1 e 100 e frequência acima 8 (DNA Nipponbare)

Software	**Tempo(s)**	**Número de TR**
ConvolutionTR	178	60982
Mreps	17	23791
Phobos	331	59619
TRF	692	31396
Swan	14112	9800

Tabela 6 – Tempo de execução para se obter todos os TR de periodo entre 2 e 100 e frequência acima 8 (DNA Arabdopsis thaliana)

Software	**Tempo(s)**	**Número de TR**
Phobos	261	3739
Swan	670	1098
ConvolutionTR	173	3777
TRF	692	649
Mreps	17	2345

Com os resultados das tabelas 1, 2 e 3, mostrados na secção anterior, pode-se perceber que a nova proposta é superior tanto em tempo quanto de execução quanto em termos de se obter TR. Isto ocorre pelo fato do algoritmo somente necessitar de comparar a seqüência lida com ela mesma com um dado atraso k, para se obter TR de período k, tal como foi mencionado na seção "Method". Esta comparação leva a se obter todos os tandem possíveis para o período (atraso k) em questão. Esse é o grande mérito desta proposta. Possivelmente podem existir propostas que tenham um tempo de execução menor, mas em termos de qualidade de resultado, a proposta terá todos os tandem possíveis na seqüência.

Os algoritmos comparados têm outras considerações quanto ao funcionamento para se obter TR de uma seqüência. Considerações estatísticas (TRF, SWAN) levam um tempo para serem computadas (mais instruções computacionais) e não conseguem obter todos os TR mas conseguem obter um razoável número de TR, porém em tempo considerável. Phobos foi o melhor destes, e teve um bom desempenho comparativo. Porém, devido a simplicidade da nova proposta, o que implica em menos instruções computacionais para serem realizadas, não teve um desempenho que pudesse ser melhor que a nova proposta. Mreps teve um ótimo desempenho em termos de tempo mas obteve menos de 1% de TR que a nova proposta ou Phobos tiveram.

CONCLUSÃO

A nova proposta apresentada possui um desempenho muito bom, quando comparados a algumas propostas existentes na literatura. É relativamente simples de implementar e pode ser usada para se obter qualquer tipo de seqüência repetitiva, não importa o quão complexa e quão grande esta seja.

REFERÊNCIAS

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990) "Basic local alignment search tool", J. Mol. Biol., 215, 403-410;

Bannert, N. & Kurth, R. (2004) "Retroelements and the human genome:

new perspectives on an old relation", PNAS, 101, 14572-79;
Botstein, D., White, R. L., Skolnick, M. and Davis, R.W (1980) "Construction of Genetic-Linkage map in man using restriction fragment length", American Journal of Human Genetics, 32, 314-331;
Freeman, J. L., Perry, G. H., Feuk, L., Redon, R., McCarrol, S. A., Altshuler, D. M., Aburatani, H., Jones, K. W., Tyler-Smith, C., Hurles, M. E., Carter, N. P., Scherer, S. W., Lee, C. (2007) "Copy number variation: New insights in genome diversity", Genome Res., 16, 949-961;
Ivics, Z. and Izsvak, Z. (2006) "Transposons for gene therapy!", Current Gene Therapy, 6, 593-607;
Kazazian Jr., H. H. (2004) "Móbile Elements: Drivers of Genome Evolution", Science, 303, 1626-1632;
Jiang, T., Kearney, P. and Li, M. (2000) "Some open problems in computational molecular biology", J. of Algorithms, 34, 194-201;
Kazazian Jr., H. H. (2004) "Móbile Elements: Drivers of Genome Evolution", Science, 303, 1626-1632;
Katti, M. V., Ranjekar, P. K. and Gupta, V. S. (2001) "Differential Distribution of Simple Repeats in Eukaryotic Genomes Sequences" Mol. Biol. Evol., 18, 1161-1167;
Kim, D. S., Huh, J. W. and Kim, H. S. (2007) "Transposable elements in human cancers by genome-wide EST alignment", Genes & Genetic System, 82, 145-156;
Kurtz, S., Choudhuri, J. V., Ohlebusch, E., Schleiermacher, C., Stoye, J. and Giegerich, R. (2001) "REPuter: The Manifold Applications of Repeat Analysis on Genomic Scale", Nucleic Acids Research, 29, 4633-4642;
Lafreniere, R. G., Brown, C. J., Rider, S., Chelly, J., Taillon-Miller, P., Chinault, A. C., Monaco, A. P. and Willard, H. F. (1993) "2.6 Mb YAC contig of the human X inactivation center region in Xq13:Physical linkage of the RPS4X, PHKA1, XIST and DXS128E genes", Hum. Mol. Genet., 2, 1105-1115;
Lander E. S. and Botstein D. (1989) "Mapping Mendelian Factors Underlying Quantitative Traits Using RFLP Linkage Maps", Genetics, 121, 185-199;

McClintock, B. (1950) "The origin and behavior of mutable loci in maize", PNAS, 36, 344-355;
Pevzner, P. and Tesler, G. (2002) "Genome Rearrangements in Mammalian Evolution: Lessons from Human and Mouse Genomes", Genome Res., 13, 37-45;
Price, A. L., Jones, N. C. and Pevzner, P. A. (2005) "De novo identification of repeat families in large genomes", Bioinformatics: 21, Suppl 1, i351-i358;
Rocheta, M., Dionisio, F. M., Fonseca, L., Pires, A. M. (2007) "Paternity analusis in excel", Computer Methods and Programs in Biomedicine, 88, 234-238;
SanMiguel, P., Tikhonov, A., Jin, Y. K., Motchoulskaia, N., Zakharov, D., MelakeBerhan, A., Springer, P. S., Edwards, K. J., Lee, M., Avramova Z. and Bennetzen, J. L. (1996) "Nested retrotransposons in the intergenic regions of the maize genome", Science, 274, 765-768;
Sharma, P. C., Grover, A., Kahl, G. (2007) "Mining microsatellites in eukaryotic genomes", Trends in Biotecnology, 25, 490-498;
TANDEM. Site disponível em http://nar.oxfordjournals.org/content/35/suppl_1/D80.full?ijkey=6fvSXFrtjnLj36M&keytype=ref. Site visitado em 02/05/2011.
Xin, B. and Liu, L. F. (1996) "DNA rearrangements mediated by inverted repeats", PNAS, 93, 819-823;
Wang, L. and Jiang, T. (1994) "On

the complexity of multiple sequence alignment", J. Comput. Biol., 1, 337-348;
Warburton, P. E., Giordano, J., Cheung, F., Gelfand, Y. and Benson, G. (2004) "Inverted Repeat Structure of Human Genome: The X-Chromosome Contains a Preponderance of Large, Highly Homologous Inverted Repeats that Contains Testes Genes", Genome Research, 14, 1861-1869;
Waterman, M. S. and Smith, T. F. (1981) "The identification of common molecular subsequences", J. Mol. Biol., 147, 195-197;
Zhi D., Raphael B. J., Price A.L., Tang H., Pevzner P. A. (2006) "Identifying repeat domains in large genomes", Genome Biol:

VARIAÇÃO ESTOMÁTICA EM MICRORREGIÕES DA FOLHA DE *Hancornia Speciosa* GOMES

Costa, V.B.S.[1]; Maciel, V.E.O.[1]; Almeida, G.M.A.[1]; Chagas, M.G.S.[2]; Lucena, I.[1]; Silva, M.D.[2]; Pimentel, R.M.M.[1]

vanessabastos_simoes@hotmail.com

[1]Universidade Federal Rural de Pernambuco, CNPq; [2]Universidade Federal de Pernambuco

RESUMO

Os estômatos são as células responsáveis pelas trocas gasosas nos vegetais. *Hancornia speciosa* Gomes é uma espécie brasileira que ocorre, predominantemente, em regiões com clima seco, como o Nordeste e Norte do país. Indivíduos desta espécie são encontrados em regiões costeiras, nas restingas, considerado um ambiente com características adversas para os vegetais. O objetivo deste estudo foi verificar a variação estomática em regiões distintas da folha de *Hancornia speciosa*. Folhas de indivíduos de *H. speciosa* foram coletadas em fragmento de restinga, litoral sul de Pernambuco. Fragmentos das porções mediana, ápice e base das folhas foram dissociados em NaCl, para confecção de lâminas semipermanentes da epiderme seguindo metodologia usual em anatomia vegetal. As análises foram realizadas através de imagens digitais. Nestas imagens foram realizadas as análises de densidade, número e índice estomático, através de programa de análise de imagens. Teste de variância (ANOVA) foi realizado para identificar possíveis variações nos parâmetros analisados entre as porções da lâmina foliar. Analisando folhas distintas, pode-se verificar variância significativa no número de estômatos e na densidade estomática, entretanto, não houve variância significativa entre as diferentes porções de uma mesma folha. Esta variação entre as folhas demonstra a elevada plasticidade da espécie.

Palavras-chave: Epiderme; Densidade Estomática; Restinga

INTRODUÇÃO

Hancornia speciosa Gomes, popularmente conhecida como mangabeira, é uma espécie da família Apocynaceae, caracterizada por apresentar folhas simples, alternas e opostas, com flores hermafroditas, e frutos arredondados aromáticos utilizados na alimentação humana. Nativa do Brasil, *H. speciosa* é característica do clima seco do Nordeste e Norte do país, atingindo também regiões de Cerrado que apresentam