IITM-CS6024 : Algorithmic Approaches to Computational Biology Date: Oct 4

Assignment 4 Due Date : Oct 12, 23:55

Roll No: BE16B002 Name: Anoubhav Agarwaal
Collaborators : Rahul Nikam

1. (5 points) [BWT]: Let BWT and BWM refer to the Burrows-Wheeler Transform and Matrix respectively.

   (a) (1 point) Reconstruct the string S whose BWT(S) = $nco\$toovican$.

   Solution: The string is convocation\$. The lexicographically sorted BWM(S) is given by:

   $$\begin{aligned}
   &\$convocatio\mathbf{n}\\
   &ation\$convo\mathbf{c}\\
   &cation\$convo\mathbf{o}\\
   &convocation\mathbf{\$}\\
   &ion\$convoca\mathbf{t}\\
   &n\$convocati\mathbf{o}\\
   &nvocation\$co\mathbf{o}\\
   &ocation\$con\mathbf{v}\\
   &on\$convocat\mathbf{i}\\
   &onvocation\$\mathbf{c}\\
   &tion\$convoc\mathbf{a}\\
   &vocation\$co\mathbf{n}
   \end{aligned}$$

   The bold letters represent the BWT(S).

   (b) (2 points) Find a string T distinct from the above string S such that BWM(T) agrees with BWM(S) on its 2^{nd} column (and possibly other columns).

   Solution: For the BWM(T) to agree with the BWM(S) on its 2^{nd} column, the set of 2-mers of the strings S and T must be the same. For it to agree on the first k-columns, the k-mers of both the strings must be the same.

   The string T which has the same 2-mers as S can be obtained by finding the eulerian path in the De Bruijn graph constructed using the 2-mers of convocation\$, i.e., S. The De Bruijn graph (in the form of an adjacency list) is given

by:

$$a->t$$
$$c->a,o$$
$$i->o$$
$$n->\$,v$$
$$o->c,n,n$$
$$t->i$$
$$v->o$$

The only other possible eulerian path (It must start and end with nodes of odd degree) in the graph is catioconvon\$. The lexicographically sorted BWM(T = catioconvon\$) is given by:

\$c*atioconvon*
at*ioconvon*\$*c*
ca*tioconvon*\$
co*nvon*\$*catio*
io*convon*\$*cat*
n\$*catioconvo*
nv*on*\$*catioco*
oc*onvon*\$*cati*
on\$*catioconv*
on*von*\$*catioc*
tio*convon*\$*ca*
vo*n*\$*catiocon*

We observe that the BWM(S) and BWM(T) agree on the first two columns (compare with (a)).

(c) (2 points) Find a string T' distinct from the above string S such that BWM(T') agrees with BWM(S) on its 2^{nd} and 3^{rd} columns (and possibly other columns).

Solution: The 3-mer composition of the strings T' and S need to be the same. Such a string can be obtained by finding the eulerian path in the De Bruijn graph constructed using the 3-mers of convocation\$, i.e., string S.

The De Bruijn graph (in the form of an adjacency list) is given by:

$$at -> ti$$
$$ca -> at$$
$$co -> on$$
$$io -> on$$
$$nv -> vo$$
$$oc -> ca$$
$$on -> n\$, nv$$
$$ti -> io$$
$$vo -> oc$$

We observe that only one Eulerian path is possible in this graph, which corresponds to string S. Thus, no distinct string T' exists such that BWM(T') agrees with BWM(S) on its 2^{nd} and 3^{rd} columns

2. (5 points) [ST, SA and Substrings]: Let SA and ST refer to suffix array and suffix tree respectively.

   (a) (1 point) Briefly describe an algorithm that uses ST to find the longest substring shared by two strings.
   (Bonus: How about a ST-based algorithm for finding the shortest substring of one string that does not appear in another string?)

Solution: Let M and N be the two given strings of length m and n respectively. Brief Algorithm:

1. Build a suffix tree of a new combined string $M\#N\$$. Here, # and \$ are unique terminal symbols. This takes $O(m+n)$ time.
2. Annotate each internal node in the tree with either M, N, or MN. The internal node is annotated as M if it only consists of leaf nodes from string M. Similarly for N. The internal node is annotated MN if it has leaves from both the strings. This takes $O(m+n)$ time using DFS.
3. Run a DFS over the tree to find the deepest internal node labeled as MN. This is also in linear time.

   (b) (2 points) A linear-time algorithm to transform the SA of a n-length string to its ST is possible with an auxiliary LCP array, which stores the length of the longest common prefix (lcp) of all pairs of consecutive suffixes in a sorted suffix array. That is, for $i = 1 \ldots n$, LCP[i] = length(lcp(suffix starting at position SA[i-1] in string,

suffix starting at position SA[i] in string)). Can you design an $\mathcal{O}(n)$ algorithm to construct the LCP array of a string from its SA?

Solution: Let SA be the suffix array of text T. Let's define the inverse suffix array SA^{-1} which is the lexicographic rank of the suffix T_j, i.e., $SA^{-1}[SA[i]] = i$ for all $i \in [0..n]$.
For LCP construction in linear time, we use the following **Lemma**:
For any $i \in [0..n)$, $LCP[SA^{-}1[i]] \geq LCP[SA^{-}1[i-1]] - 1$

Problem: LCP array construction
Input: text $T[0..n]$, suffix array $SA[0..n]$, inverse suffix array $SA^{-}1[0..n]$
Output: LCP array $LCP[1..n]$

1. $l \leftarrow 0$
2. for $i \leftarrow 0$ to $n-1$ do
3. $\quad k \leftarrow SA^{-}1[i]$
4. $\quad j \leftarrow SA[k-1]$
5. $\quad$ while $T[i+l] = T[j+l]$ do $l \leftarrow l+1$
6. $\quad LCP[k] \leftarrow l$
7. $\quad$ if $l > 0$ then $l \leftarrow l-1$
8. return LCP

Proof that run-time is $O(n)$:
(1) All steps other than 5 clearly take either constant or linear time.
(2) Each round in the loop increments l. Since l is decremented at most n times on line 7 and cannot grow larger than n, the loop is executed $O(n)$ times in total.[1]

(c) (2 points) The DC3 algorithm for SA construction (recursively) sorts suffixes starting at positions that are **not** multiples of 3 in the first step, and uses it to sort/merge suffixes that are at multiples of 3 in later steps. What will happen if you extend this idea to get a DC2 algorithm (which sorts odd-position suffixes in the first step, and uses it to sort/merge even-position suffixes in the next steps)? That is, will the DC2 algorithm steps be "easier or trickier" to implement than the DC3 algorithm steps? Justify briefly.
(Bonus: What subset of suffixes will you sort in the first step of a DC7 algorithm to get an algorithm similar to the DC3 algorithm?)

Solution: The DC2 algorithm will be trickier to implement compared to the DC3 algorithm in the later steps.

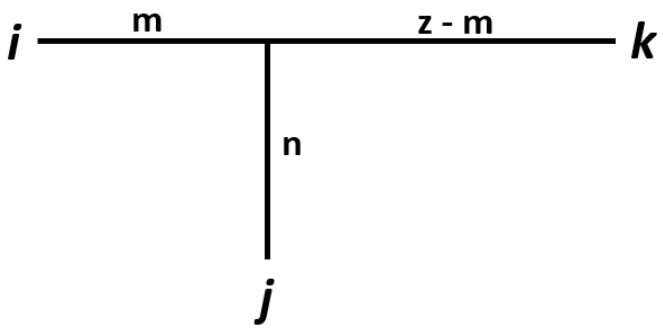
3. (10 points) [Coding Time]: Neighbor-Joining (NJ) is a quite popular bioinformatics algorithm with 30,000+ citations making it one of the top 20 most cited papers in all fields of science!

   (a) (8 points) Solve the contest at https://www.hackerrank.com/cs6024assignment-4 by implementing NJ to reconstruct a phylogenetic tree from a distance matrix. Please follow the instructions given carefully to produce the exact same output as the test cases.

   (b) (2 points) In your algorithm, you could've used n=3 instead of n=2 leaves as base case in recursion. Related to this, prove that every 3×3 distance matrix is additive.

Solution: Let $d_{ij}(T)$ represent the distance between i and j in the tree T. Let D be a 3×3 distance matrix for 3 species i, j, and k such that $D_{ij} = x$, $D_{jk} = y$, $D_{ki} = z$. Here, D_{ij} represent entries in the distance matrix.

To Prove: Every 3x3 matrix is additive, i.e., there exists a tree T with $d_{xy}(T) = D_{xy} \forall\ x, y$ (1)



(Refer to figure) WLOG, Connect the nodes of the tree i and k with an edge of length D_{ik}, i.e., $d_{ik} = D_{ik} = z$. Take any point at a distance of m from i on this edge. From this point let species j be at a distance of n.

For condition (1) to hold and thus the a general 3×3 matrix to be additive, the following equations need to be satisfied:

$$\begin{aligned} d_{ij} &= D_{ij} = x \\ d_{ij} &= m + n = x \qquad (i) \\ d_{jk} &= D_{jk} = y \\ d_{jk} &= z - m + n = y \qquad (ii) \end{aligned}$$

Equations (i) and (ii) are a system of two linear equations with two unknowns m and n. The equations are always solvable. The solution is give by:

$$\begin{aligned} m &= (x + z - y)/2 \\ n &= (x - z + y)/2 \end{aligned}$$

Hence, for any 3×3 distance matrix, we can obtain its corresponding tree.

4. (10 points) [Research critique]: Please provide critique on the research paper assigned to you (students are randomly assigned another team's paper to critique; please find your paper from the allotment file here).

   (a) (3 points) Please provide a brief overview of the paper in the format described. (Format: Mention the (i) biological question addressed in the paper, (ii) computational problem (in the same format as in the textbook/lecture-slides) studied in the paper, and (iii) computational methods/approach taken to solve the problem.)

Solution: Paper Title: Novel risk genes for systemic lupus erythematosus predicted by random forest classification.
(i) **Biological question**
Systemic lupus erythematosus is an autoimmune disease whereby the immune system of the body attacks its healthy cells. The disease is characterized by inflammation of organs. There is no cure available to treat SLE completely. Treatment options involve bettering quality of life and alleviating symptoms. As SLE is a genetic disease, people can be at a predetermined risk of getting the disease. To identify the genetic components that play a part in the disease and to identify novel genes that are implicated in SLE, the study uses machine learning algorithms for better prediction. To further validate the data, the expression patterns of B-cells and T-cells are studied for the identified genes, and their regulatory mechanisms are investigated to study cell-dependent genetic contributions.

(ii) **Computational problem**

1. Random Forest Classifier
2. Logistic Regression

The paper investigates the difference in prediction accuracy in the above-mentioned models for the same datasets. These two models are also used to predict the risk for Lupus Nephritis, a severe manifestation of SLE.

Input: SNP genotype data
Output: Gini importance score and Probability value

(iii) **Computational method**
The paper implements a Random Forest classifier to predict SLE risk and identify genes that might be involved in conferring SLE risk in individuals.
R package used: EMIL (Evaluation of Modeling without Information Leakage)
The status prediction was performed with 15 fold cross-validation runs involving 3000 SNPs per classification fold.

(b) (3 points) Mention three strengths or key contributions of the paper in your own words (not copied as is from paper).

Solution: The key contributions of the paper are:

1. The paper designed a random forest classifier to predict whether a person is at risk of SLE. Random forest was able to give a better prediction accuracy, with an AUC score of 0.78. Furthermore, they were able to give better prediction accuracy when a severe form of SLE, lupus nephritis, was used as a test case.

2. The investigating team genotyped many individuals from Sweden and collected their SNP information. This genotyped data was comparable with the GWAS catalog in terms of the predicted genes.

3. The classifier was able to identify 3 novel genes to be implicated in SLE. These 3 genes haven't been previously indicated in SLE. The genes have a high gene importance score denoted by RF classifier. Moreover, these genes rank alongside 37 other genes, some of which have been implicated in SLE by GWAS studies. Similarly, they were able to identify 6 novel genes implicated only in Lupus Nephritis.

4. After identifying genes that may play a role in SLE, they studied the expression patterns of the genes. This study further helps validate the functional roles of the novel genes. They furthermore studied whether B-cells or T-cells contain the overexpressed genes and studied the gene expression preference between the two. This analysis not only helps understand the expression of genes but also highlights the importance of cell-specific expression analysis to understand the disease better. Furthermore, the paper also investigates the transcriptional regulation mechanism of the top 40 genes. Hence the paper has firmly established their findings with genetic and gene expression findings along with information about the cis-regulatory elements.

(c) (3 points) Mention three areas of further improvement of the paper.
(Note: If it is a theoretical paper, state whether the proofs were easy to read, if there was clarity in the presentation, etc. It there is empirical investigation, you can comment about the results obtained, if those results mentioned can be replicated or improved using some other idea you have. You can also write about the gaps in the paper, and ideas you may've about filling those gaps.)

Solution: Areas for further improvement of the paper:

1. While random forest for SLE gives a prediction accuracy of 0.78, which is an improvement over logistic regression, it hasn't been mentioned why

this difference exists. Moreover, SLE lupus nephritis gives an AUC score of 0.91, which is the highest score obtained. No explanation as to why this result gives a better prediction accuracy only for the RF classifier has been given. Further enhancements to improve the SLE prediction accuracy have not been explored.

2. Comparisons with more models like Bayes classifier will help understand why/why not RF classifier works best. No previous work has been cited for the same.

3. The study investigates the cell-specific regulation and preferences of gene expression. In the study, the top 40 genes show a significant over-expression in T-cells compared to B-cells. Moreover, there are more B-cell specific genes than T-cell specific genes in the top 40 identified genes. The reason for this hasn't been explored. In an autoimmune disease, there are more immune cells that play an essential part in carrying out an immune attack. Why these specific identified genes show enriched B-cell expression and not T cell needs to be further explored. Along with this, the genes expressed in T-cell and associated with SLE need to be identified. Similar studies should be extended to Lupus Nephritis for functional validation.

(d) (1 point) Considering both above, please mention whether the work is a significant contribution to this field.

Solution: The work conducted is significant as it lays the groundwork for the use of machine learning techniques for a complex disease like SLE. It also investigates cell-specific gene expression as well as the genetic factors of the disease. However, more work is required to identify more genes and improve the prediction of risk status for other demographics, with the use of other genotyping and machine learning tools.

References

[1] https://www.cs.helsinki.fi/u/tpkarkka/opetus/11s/spa/lecture10.pdf