Developing the Performance of Tiling Arrays

Mohamed Abdelhamid Abbas, King Khalid University, Saudi Arabia

ABSTRACT

Genomic tiling arrays are able to inspect the genome of haphazard species for which the sequence is known. The plan of proper oligonucleotide probes for such arrays is computationally difficult if features such as oligonucleotide quality and recurring regions are considered. Prior works have developed the minimal tiling path problem for the choice of oligonucleotides using Dijkstra’s shortest path algorithm to compute universal finest tiling paths from millions of candidate oligonucleotides on computers. Although Dijkstra’s algorithm works well, it is complicated and may take a long time for routers to process it and the efficiency of the network fails. In this paper, the author discusses a search approach that can decrease the average complexity time of tiling arrays. This aspiration is realized by searching for the shortest path to the probes using a faster algorithm. This paper enhances A* Algorithm and exploits the enhanced version, called A**, instead of Dijkstra’s algorithm. The enhanced version is more efficient and can decrease the average time complexity, thus increasing the performance of tiling array.

Keywords: A* Algorithm, Dijkstra’s Algorithm, DNA Microarrays, Oligonucleotide Probes, Tiling Arrays

1. INTRODUCTION

Tiling Arrays hybridize labeled target molecules to unlabeled probes fixed on to a solid surface. Tiling arrays vary in the character of the probes. Number of features on a single array can range from 10,000 to greater than 6,000,000, with each feature containing millions of copies of one probe (Schliep, A., & Krause, 2007; Kane, Jatkoe, Stumpf, Lu, Thomas, & Madore, 2000; Matveeva, Shabalina, Nemtsov, Tsodikov, Gesteland, & Atkins, 2003). Tiling arrays can produce an unbiased look at gene expression because previously unidentified genes can still be incorporated. They are quickly becoming one of the most powerful tools in genome-wide investigations. It is possible to use multiple probes per gene to measure its expression level. Representing a genome completely opens new routes, such as transcription profiles that do not rely on prior gene prediction. An oligonucleotide is a short nucleic acid polymer, typically with twenty or fewer bases (He, Wu, Li, Fields, & Zhou, 2004).

Although they can be formed by bond cleavage of longer segments, they are now more commonly synthesized by polymerizing individual nucleotide precursors. Automated synthesizers allow the synthesis of oligonucleotides up to 160 to 200 bases (Pozhitkov et al., 2006; Wikipedia, n. d.; Hoogenboom, Fijten, & Schubert, 2002). An Oligonucleotide probe

DOI: 10.4018/jcmam.2011070102
is a short sequence of nucleotides synthesized to match a region where a mutation is known to occur and then used as a molecular probe to detect the mutation (MedicineNet, n. d.). Alberts et al. (2007) present an extensive and generalized protocol for the verification of probe sequences. The determination of suitable probe sequences, typically contiguous subsequences of the genomic sequence, however is computationally demanding even for relatively small genome sizes of bacteria if the quality of probes is considered in the design (Wu & Irizarry, 2004). The probe selection for the tiling array needs to be customized, which is economically feasible as several providers offer customer designed arrays.

The determination of suitable probe sequences is computationally demanding if the quality of probes is considered in the design. The multi-criterion optimization problem of selecting an optimal subset of oligonucleotide probes is formulated from set of candidates as the minimal cost tiling path problem. This problem is equivalent to a shortest path problem, which is solved in prior works using Dijkstra’s shortest path algorithm (Schliep & Krause, 2007). Prior work depends to find the shortest path using Dijkstra’s shortest path algorithm (Schliep & Krause, 2007; Dai, 2005). Another algorithm that find the shortest path is A* algorithm. It is an efficient more than Dijkstra’s algorithm but its applications appears only in engineering science such as civil engineering, town planning and roads. The next subsections depict the structure and the behavior of these algorithms in details (Dai, 2005).

2. PRIOR WORKS

Prior works formulated the problem of finding the shortest path between the start oligonucleotide probe and the destination one as finding the shortest path in a graph as depicted in Figure 1. The set of vertices are the probes \{v1, . . . , vn\} with special nodes 0 and n + 1, which are virtual probes. It is required to tile before the start and after the end of the sequence. For an edge (va, vb), with va≥1, vb ≤ n Where n is the total number of vertices in the graph. The total weight is computed as w(va, vb). The weights w(0, va) and w(va, n+1) are defined as d(0, va). Prior work depends to find the shortest path using Dijkstra’s shortest path algorithm (Schliep & Krause, 2007; Dai, 2005).

2.1. The Dijkstra’s Shortest Path Algorithm

The Dijkstra’s shortest path algorithm is the most commonly used to solve the single source shortest path problem today. For a graph G(V, E), where V is the set of vertices and E is the set of edges, the running time for finding a path between two vertices varies when different data structure are used. The run time of first for loop is O(V). In each iteration of the while loop the total run time is O(E). Figure 2 depicts the steps of algorithm in details (Dai, 2005; Chou, Romeijnz, & Smith, 1998; Engineer, 2001; Shad, Ebadi, & Ghods, 2008; Fu, Li, & Deng, 2004; Cormen, Stein, Rivest, & Leiserson, 2001; UCD School of Computer Science and Informatics, n. d.).

2.2. A* Shortest Path Algorithm

A* is a graph search algorithm that finds a path from a given initial node to a given goal node. It employs a “heuristic estimate” that ranks each node by an estimate of the best route that goes through that node. It visits the nodes in order
Metaheuristic Optimization of Constrained Large Portfolios using Hybrid Particle Swarm Optimization
www.igi-global.com/article/metaheuristic-optimization-of-constrained-large-portfolios-using-hybrid-particle-swarm-optimization/171460?camid=4v1a