HCV Quasispecies Assembly Using Network Flows

Kelly Westbrooks[a⋆], Irina Astrovskaya[a⋆], David Campo Rendon[b], Yury Khudiakov[b], Piotr Berman[c], and Alex Zelikovsky[a]

[a]Department of Computer Science, Georgia State University, Atlanta, GA 30303
{csckew,iraa,alexz}@cs.gsu.edu
[b]Centers for Disease Control and Prevention, Atlanta, GA 30333, {fyv6,yek0}@cdc.gov
[c]Department of Computer Science and Engineering, Pennsylvania State University
University Park, PA 16802, berman@cse.psu.edu

Abstract. Understanding how the genomes of viruses mutate and evolve within infected individuals is critically important in epidemiology. By exploiting knowledge of the forces that guide viral microevolution, researchers can design drugs and treatments that are effective against newly evolved strains. Therefore, it is critical to develop a method for typing the genomes of all of the variants of a virus (quasispecies) inside an infected individual cell.

In this paper, we focus on sequence assembly of Hepatitis C Virus (HCV) based on 454 Lifesciences system that produces around 250K reads each 100-400 base long. We introduce several formulations of the quasispecies assembly problem and a measure of the assembly quality. We also propose a novel scalable assembling method for quasispecies based on a novel network flow formulation. Finally, we report the results of assembling 44 quasispecies from 1700 bp long conserved region of HCV.

1 Introduction

Many viruses found in nature encode their genomes in RNA rather than DNA. While the problem of sequencing an organism's DNA is well-studied, sequencing RNA viruses presents its own unique set of challenges. Perhaps the biggest challenge associated with sequencing RNA viruses is that they lack DNA polymerases and are unable to repair mistakes in their sequences as they reproduce. Over the course of infection, the mistakes made in replication are passed down to descendants, producing a family of related variants of the original viral genome referred as a *quasispecies*.

The allele frequencies across all of the quasispecies in an infected individual may drift significantly. Among all of the new quasispecies produced, some may be more virulent than others. Thus, it is of epidemiological interest to identify common characteristics of virulent quasispecies to aid in the design of effective drugs and treatments for the disease that the virus causes. This paper is devoted to the problem of sequencing of all quasispecies inside a patient based on 454 Lifesciences system.

454 Lifesciences system is one promising technology that may prove useful for sequencing quasispecies. It is a massively-parallel pyrosequencing system developed

⋆ Partially supported by GSU Molecular Basis of Disease Fellowship.

by biotechnology firm 454 Lifesciences for DNA sequencing. Briefly, the system fragments the source genetic material to be sequenced into pieces approximately 100 bp long called *reads*. Each read is sequenced and the original genome is reconstructed via software. Since this system was originally designed to sequence genetic material from a single organism, the software assembles all of the reads to a single genome. In order to use it for sequencing quasispecies, new software must be created that can also correctly distribute reads between multiple quasispecies.

Informally, the Quasispecies Assembly problem can be stated as follows: Given a set of reads taken from a single specimen, determine how many quasispecies are present and what are their sequences.

Quasispecies Assembly is related to several well-known problems: DNA fragment assembly (see e.g., [4, 5]), haplotype assembly [2] and population phasing (see e.g., [6]). Indeed, the fragments (reads) should be assembled into a long genome sequence although it becomes a lesser challenge since consensus genome sequence is already available. A plausible reduction genome sequencing is as follows: place all quasispecies genomes back-to-back in a long sequence and treat common segments as repeats. Quasispecies Assembly is very close to the haplotype assembly problem where fragments are given from two different haplotypes of the same diploid organism and the goal is to correctly attribute segments to one of these two haplotypes. Unfortunately, the proposed solution methods are not scalable with respect to the number of haplotypes per individual and this is critical since in a specimen there are hundreds or even thousands of different quasispecies. Therefore, one can find similarity with the population phasing problem where multiple diplotypes (mixtures of two haplotypes) are given and it is required to identify underlying common haplotypes and their frequencies. Finally, it can also be viewed as variant of the newly-arisen problem of finding and distinguishing all the different species in a given DNA sample – but in our case, the complicating factor is that the sequencing of different quasispecies are very similar to each other.

Our contributions are the following:

- Several optimization formulations for Quasispecies Assembly and its different versions.
- Estimation of the probability that two overlapping read belong to the same quasispecies.
- A network flow based method for solving the quasispecies assembly problems.
- An efficient and scalable implementation of the proposed network flow methods.
- Application of the network flow method to the set of simulated reads drawn from 44 quasispecies in the E1E2 region of Hepatitis C Virus.

In the next section we give several optimization formulations for Quasispecies Assembly. Then we will construct proposed data structure incorporating information about given reads and the consensus genome. Section 4 will propose solutions for Quasispecies Assemble problems based on reductions to network flows. Finally, Section 5 will describe validation of network flow approaches on E1E2 region of HCV.

2 Quasispecies Assembly Models and Optimization Formulations

The ultimate goal of Quasispecies Assembly is to correctly reconstruct genomes of all quasispecies in a given sample. Since multiple quasispecies have indistinguishable common segments that are significantly longer than a read, one cannot guarantee to find even the exact number of quasispecies. Although only cross-validation of proposed techniques can really tell if their quality are of practical interest, it is important to formulate models and corresponding optimization objectives that do not simply rely on cross-validation.

We will start with the formal description of the input and output for Quasispecies Assembly. The output of 454 Lifescience system consists of $N \approx 250K$ reads, each read r is a sequence of l nucleotides (l may be about 100 or even 400). The rate of typing errors is claimed to be 0.04% (see [8]). Also we may usually rely on existence of a known consensus genome of all quasispecies which is in case of HCV has length $L = 9600$ bp. Each reconstructed quasispecies should be covered by given reads and be close to the consensus genome sequence H.

We first consider the simplest parsimonious model for Quasispecies Assembly. The corresponding optimization formulation is as follows.

Most Parsimonious Quasispecies Assembly. Given a set of reads R and a consensus genome sequence H, find the minimum size set of quasispecies Q covering all reads from R, i.e., such that each read $r \in R$ is contained in at least one $q \in Q$.

Although the parsimonious model is worth considering, it is usually oversimplified. Indeed, it usually predicts less than observed number of quasispecies and cannot distinguish between numerous different equally good (from parsimonious point of view) solutions. In order to break ties, we introduce penalties over read overlaps. The penalty $cost(r, r')$ over an overlap between reads $r, r' \in R$ should reflect how unsure we are that these two reads came from the same quasispecies. We set $cost(q)$ to be the sum of costs of constituting overlaps. For example, cost can be inversely proportional to the probability that such overlap occurs. Then the overall probability of the quasispecies q is the product of costs of consecutive overlapping pairs of reads which can be transformed to the sum by replacing costs with their logarithms. In the next section we will suggest several different cost functions.

Minimum Cost Parsimonious Quasispecies Assembly. Given a set of reads R with costs on read overlaps and a consensus genome sequence H, find the most parsimonious set of quasispecies Q that have the total minimum cost.

We may also trade the number of quasispecies for smaller cost or just completely disregard minimization of the number of quasispecies.

Minimum Cost Quasispecies Assembly. Given a set of reads R with costs on read overlaps and a consensus genome sequence H, find the set of quasispecies Q covering all reads that have the total minimum cost.

Besides accurately assembling all quasispecies as a set, it is also important to assemble certain frequent individual quasispecies. There is an evidence that the frequency distribution of quasispecies in a single cell is usually not uniform. The most frequent

quasispecies may contribute the major part of reads and may also contribute the most to virus resiliency. Although frequently repeated reads may come from the most frequent quasispecies, the alternative explanation would be that multiple different quasispecies have the same common segment. Therefore, we again rely on estimated probability for overlapping reads. This results in the following problem formulation.

The Most Frequent Quasispecies Assembly. Given a set of reads R with costs on read overlaps and a consensus genome sequence H, find a single quasispecies q with the minimum total cost.

3 The Read Graph

In this section we propose the method of incorporating the input information about reads and genome consensus sequence into a single data structure to which we apply network flow methods for solving quasispecies assembly problems.

We will first describe how to align reads to the consensus genome and distinguish single nucleotide polymorphisms (SNP) from typing errors. For every possible starting position, we align both the read and its reverse compliment to the consensus sequence and count the number of mismatches. The "true" starting position has the fewest number of mismatches. In our experiments we have never encountered read misalignments which can be explained by a lack of sizable repeats in the RNA viral genomes and the low typing error rate (0.04% [8]). We can distinguish typing errors from infrequent SNPs if we have at least double coverage of each quasispecies – indeed, the probability of the same typing error occurring twice is insignificantly small.

Formally, each read r is supplied with its beginning and ending positions in the consensus sequence b_r and e_r, respectively. The *read graph* $G = (V, E, cost)$ has the set of vertices V each representing a read and the set of directed edges E, where each edge (u, v) connects two reads u and v if their alignments overlap (i.e., $b_u \leq b_v \leq e_u \leq e_v$) and if they coincide with each other across the overlapping region.

Obviously, some edges correspond to *true* overlaps of pairs of reads coming from the same quasispecies while other correspond to false overlaps that occur between reads of similar but different quasispecies. The cost function of an edge (u, v) reflects how unsure we are that u and v correspond to a true overlap.

In the next subsection we describe how we reduce the size of the read graph without losing any information. Our reduction is based on an efficient algorithm for minimum transitive reduction. In Subsection 3.2 we estimate the probability for an edge in the transitively reduced read graph to correspond to a true overlap.

3.1 Transitive Reduction of the Read Graph

In general, the read graph G may be very dense since it contains edges connecting non-consecutive reads (see Figure 3.1). If there are three reads u, v and w such that $(u, v), (v, w) \in E$ and u overlaps with w (i.e., $b_w < e_u$), then $(u, w) \in E$. The path $u - v - w$ is called *closed* since there is a single edge (u, w) connecting the beginning with the end. The edge (u, w) is logically implied by the other edges and we can safely remove it without losing any information.

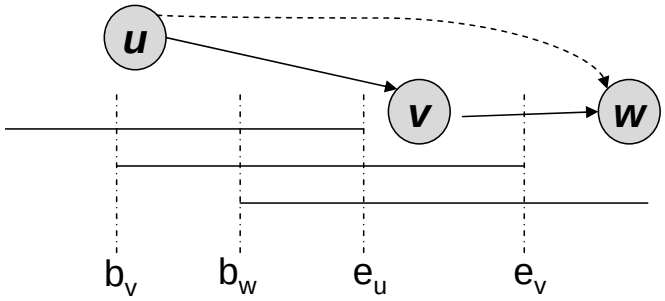

Fig. 1. The edge (u, w) is logically implied by the edges (u, v) and (v, w).Indeed, the segment $[b_w, e_w]$ is the same in the reads u and v since $(u, v) \in E$ and $[b_w, e_w]$ is the same in the reads v and w since $(v, w) \in E$, therefore, it is the same for u and w.

Thus, we wish to remove maximum possible number of edges, or, in other words, obtain the minimum transitive reduction $G' = (V, E')$ of the graph G. The *transitive reduction* is a subgraph of G in which if a vertex v is reachable from u, then it should be reachable in G'. In general, finding minimum transitive reduction is NP-complete but since G is is a directed acyclic graph, it can be found efficiently [9]. Besides lacking directed cycles, the read graph G is also *partially transitively closed*, i.e., all subpaths of closed paths are closed.

Input: Partially transitively closed directed acyclic graph $G = (V, E)$
Output: Minimum transitive reduction of G

1. Topologically sort vertices of G
2. For each vertex $u \in V$ in topological order do
3. Sort all outgoing edges from u according to left end: $v_1, \ldots, v_k$
4. Thread set $T \leftarrow \emptyset$
5. For $i = 1, \ldots, k$ do
6. For each $x \in T$ do
   If edge $(x, v_i) \in E$, then $E \leftarrow E - (u, v_i)$, $T \leftarrow T - x$, break
   $T \leftarrow T \cup v_i$
7. Output G

Fig. 2. Minimum transitive reduction for partially transitively closed directed acyclic graph.

Claim. A read graph G is partially transitively closed.

Proof. Toward contradiction assume that there exists a closed $u - v$-path P without chords. Let (w, v) be the last edge of P, we will show that there exists the edge (u, w). Indeed, existence of $u - w$-path and (w, v)-edge implies that $b_u \leq b_w \leq b_v \leq e_u$ and,

therefore, u and w overlap. Since there exists a $u - w$-path, u and w do not disagree. These two facts imply there should be an edge u and v.

The following algorithm for finding minimum transitive reduction (see Figure 2) is more efficient than for general directed acyclic graphs since it relies on G being partially transitively closed. The runtime is $O(\delta|E|)$, where δ is the maximum number of quasispecies containing the same read and $|E|$ is the number of edges in G. This is significantly faster than $O(|V|^2)$ for arbitrary directed acyclic graphs.

From now on, we assume that the read graph G is transitively reduced. Obviously, an arbitrary quasispecies corresponds to unextendable path of G, although not every unextendable path corresponds to a quasispecies.

3.2 Estimating Probability of a True Overlap

We first give intuition behind estimation of the true overlap probability and then present results of the formal analysis of the uniform and non-uniform quasispecies distributions.

Intuitively, given a choice, one would trust a larger read overlap more than a smaller read overlap. That makes a lot of sense in the standard sequencing when the consensus genome is unknown. The entire de Bruijn graph approach relies only on sufficiently long overlaps (see [3]). Indeed, it is quite improbable that a long overlap happens by chance – only repeats may result in false long read overlaps. But Quasispecies Assembly exactly the case with long and frequent repeats – many segments can be repeated in very many quasispecies.

Only multiple coverage may give a clue for deciding which overlaps are probably true. If there are two reads u and v adjacent in the transitively reduced read graph, then we may try to measure our surprise with the fact that $(u, v) \in E$ by the length of the "overhang" $\Delta = |b_v - b_u|$. Indeed, assuming that (u, v) represents a true overlap in a quasispecies q, why there is no other read w that is taken from q and which is between u and v? If Δ is large, then there great chance that the overlap is false.

Formally, let us consider a simplified model where every read has the same length and that each quasispecies has the same frequency.

Let b_u be the starting position, in sequence H, or read u. After transitive reduction, the event that two reads u, v from the same quasispecies are connected with an edge (u, v) is the event that (a) these two reads exist, and (b) no read w from the same quasispecies satisfies $b_u < b_w < b_v$.

Let us fix a quasispecies A. Given N reads, L positions in H and q quasispecies, the probability that a position is b_u for some read u of A is N/Lq. Assuming that b_u is such a position, there is a unique true edge (u, v) indicating an overlap with another read of A. The event that $b_v - b_u > k$ is the event that $b_u + 1, b_u + 2, \ldots, b_u + k$ are not beginnings of reads of A, and since the reads have uniformly random positions, the probability of that event is

$$p_k = \left(1 - \frac{N}{Lq}\right)^k \approx \exp(-kN/Lq)$$

The probability that $b_v - b_u = k$ is $p'_k = \frac{N}{Lq} p_{k-1}$.

If $b_v - b_u$ is much larger than Lq/N then most probably b_v is a read from another quasispecies B, and the reason for the difference is not a gap between the positions of various reads of A, but the fact that in the interval between b_u and b_v the sequences of quasispecies A and B are different. Therefore if $\Delta = b_v - b_u$, the number $1/p_\Delta \approx= \exp(\Delta N/Lq)$ measures the "implausibility" that (u, v) is a true edge. By "implausibility" we mean a quantity that is low when the edge is plausible and high when it is not.

If the lengths of the reads are variable and random, then after the cleaning we should have larger gaps following beginnings of particularly long reads. However, the distributions of lengths of the survivors of the cleaning process is more uniform (it has a much smaller variance) than the original distribution of read lengths. Thus we have a reasonable approximation.

If we have different frequencies of reads from different quasispecies, then the proper formula for quasispecies A would be $\exp(\Delta N f_A/L)$ rather than $\exp(\Delta N/Lq)$. However, We cannot use it because a-priori we do not know the frequencies.

What is least clear from our analysis is what function of p_Δ would give the best result if we use it as the cost of edge (b_u, b_v). We will try to natural candidates: the inverse, giving formula $\exp(\Delta N/Lq)$ and minus logarithm, giving formula $\Delta N/Lq$.

4 Quasispecies Assembly via Network Flows

In this section we show how to modify the read graph G into a flow network so that Quasispecies Assembly would naturally represented by a network flow through G. We then reformulate the Quasispecies Assembly problems into the minimum-cost network flow problems.

As we noticed in Section 3.1, each quasispecies corresponds to a simple path in the (transitively reduced) read graph G. Each such path can be viewed as a *flow* originated in the source corresponding to the first read flowing through intermediate reads and ending at the sink corresponding to the last read.

Standard network flow formulations associate flow with the edges rather than the vertices. Therefore, our first modification to the read graph would be replacement of each vertex corresponding to a read r with the beginning vertex r_b and the ending vertex r_e connected with the edge (r_b, r_e). Each edges with the head v changes its head to r_b and each edge with the tail v changes it tail to r_e (see Figure 3).

For simplification of the flow formulation we also introduce universal source and sink vertices s and t for all flows (see Figure 4). We add an edge from the source s to each read that does not have any incoming edges and an edge to the sink t from each read that does not have outgoing edges. These two vertices are also supplied with back edge (t, s) through which each quasispecies flow should return back thus making our flow circular.

We now ready to formulate the minimum cost feasible flow problem corresponding to Most Parsimonious Quasispecies Assembly. Let $f : E \to R_0^+$ be a *circular flow* defined on all edges. The value of the flow f through a read edge (b_r, e_r) represents the number of quasispecies that contain the read r. The corresponding linear program (1-4) is as follows:

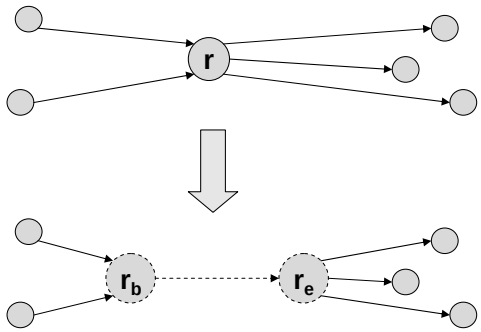

Fig. 3. Replacing of a vertex corresponding to a read r with the edge (r_b, r_e). The new vertices and edges are dashed.

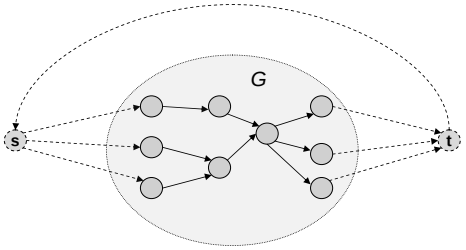

Fig. 4. Universal source s and universal sink t with the backward edge (t, s) are added to the read graph. The new vertices and edges are dashed.

$$\text{Minimize} \quad f(t,s) \tag{1}$$

subject to

$$\forall v \in V \quad \sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u) \tag{2}$$

$$\forall \text{read } r \in R \;\; f(b_r, e_r) \geq 1 \tag{3}$$

$$\forall (u,v) \in E \;\; f(u,v) \geq 0 \tag{4}$$

Objective (1) is parsimonious – it asks for minimizing number of quasispecies since each unit of flow corresponding to a single quasispecies should pass through the edge (t, s) exactly once. Constraint (2) is the flow conservation – for each vertex $v \in V$, the total flow entering v equals the total flow exiting v. Constraint (3) requires that each read to be covered by at least one predicted quasispecies. Constraint (4) forbids the backward flow so that the flow would really correspond to quasispecies.

The linear program (1-4) does not predict complete quasispecies but rather decides which pairs of overlapping reads belong to the same quasispecies. In order to obtain a feasible set of quasispecies one can simply replace each edge e with $f(e)$ copies and in the resulted graph find $f(t,s)$ edge-disjoint $s-t$-paths each corresponding to a quasispecies.

Although the linear program (1-4) does not require flow to be integer, the optimal integer solution is always fractionally optimal. All linear program solvers (e.g., [**?**,**?**]) find optimal integer solutions very efficiently. Alternatively, one can use a faster combinatorial min-cost flow solver [10, 11].

The next linear program solves Minimum Cost Quasispecies Assembly. Here, we set to zero the cost of all edges introduced into G while modifying it into the flow network, i.e., $cost(t,s)=0$, $cost(s,u)=cost(v,t)=0$ and, for each read r, $cost(r_b,r_e)=0$.[1] The only difference with the previous formulation is in the objective. Objective (5) does not pay attention to the number of predicted quasispecies but to the total cost of all predicted quasispecies.

$$\begin{aligned}
&\text{Minimize} \quad \sum_{e\in E} cost(e)\cdot f(e) && (5)\\
&\text{subject to}\\
&\forall v\in V \quad \sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u)\\
&\forall \text{read } r\in R \;\; f(b_r,e_r)\geq 1\\
&\forall (u,v)\in E \;\; f(u,v)\geq 0
\end{aligned}$$

Finally, Minimum Cost Parsimonious Quasispecies Assembly is solved with the same linear program as Minimum Cost Quasispecies Assembly. The only difference is that $cost(t,s)$ is set to a very large number. As a result any feasible assembly cannot be optimal if it uses more than the minimum possible number of quasispecies and as a secondary criteria the total cost of read overlaps is minimized.

5 Experimental results

To our knowledge, full-genome quasispecies data for HCV is currently unavailable. However, previous research has obtained the sequences of individual HCV quasispecies for several important subregions. [1] obtained quasispecies data for the E1E2 region of the HCV genome. This data is a contiguous region 1734 bp long over $Q=44$ quasispecies. We generated two simulated problem instances based upon the sequences in this data. The first instance, which we shall refer to as the "uniform" instance, assumed that the frequencies of each of the sequences in the data set were equal all equal (i.e.

[1] Alternatively, the cost of the read can be set inversely proportional to the number of copies of the read. This way the multiplicity of the read participation in assembly should correspond to its multiplicity among collection of all reads.

Instance	# of reads	# of reads after cleaning	# of overlaps	# of overlaps after transitive reduction
Uniform	44028	7810	1047340	19664
Nonuniform	44029	6097	674082	16350

Fig. 5. This table shows the original number of reads, the number of reads after "cleaning" (i.e. removing subreads), and the number of overlaps (i.e. number of edges in the read graph) for the two problem instances considered. The "uniform" problem instance consisted of 44 quasispecies of length 1734 each with equal frequency. The "nonuniform" instance consisted of the same 44 quasispecies, but one quasispecies was selected to have frequency $1/2$ and all others were given frequency $1/(2(Q-1))$.

$f(q_i) = 1/Q$ for all q_i). The second instance, which we refer to as the "nonuniform" instance, assumed that the frequency of one quasispecies was $1/2$, while all other quasispecies had equal frequency of $1/(2(Q-1))$.

We assumed that the 454 Lifesciences system would produce approximately 250K reads of length 100 across the entire 9.6K bp length of the HCV genome. Since the E1E2 region is 1.7K bp long, approximately 18% of the 250K reads (approx. 44K) reads should span the E1E2 region. Reads were generated by iteratively selecting a sequence from E1E2 at random according to that read's frequency and fragmenting it into reads which were then accumulated in a collection; the lengths of reads were generated using a normal distribution with $\mu = 100, \sigma^2 = 10$.

Once a problem instance was generated by the above procedure, we removed reads that were contained within other reads. The reason we introduce this "cleaning" phase of our algorithm is two-fold: first, any read that is subread of another cannot possibly introduce a new quasispecies, and second, the graph formed by the remaining reads is guaranteed to be acyclic, connected, and has a single global source and sink. Due to the large degree of homogeneity between quasispecies, a surprisingly large number of reads are cleaned out of the problem instance. After cleaning the problem instance of subreads, the read graph is constructed. The table on Figure 5 gives the various parameters for each of the two problem instances under consideration.

Out of the many possible overlaps between reads in the problem instance, only a small portion actually belongs to real quasispecies. From the table on Figure 6 one can see how well our min-cost flow based algorithms for Most Parsimonious Quasispecies Assembly and Minimum Cost Quasispecies Assembly predict which overlaps are true overlaps. The most parsimonious solution obtained by setting to 1 the back edge cost while other edges has cost 0 – in the table the corresponding solution is placed the row with cost function 1. We run the min-cost flow algorithm for the two problem instances under the following two different edge-cost functions. The cost function Δ equals the the difference in genome offsets of edge tail and head reads. This function is proportional to the logarithm of the estimated probability of the read overlap to be true overlap. As a result, the total cost of a path is proportional to the probability of it to be a true quasispecies. The cost function e^{Δ} is the estimated probability of the corresponding overlap to be true overlap.

Instance	Cost Function	True overlaps	Predicted overlaps	Correctly predicted overlaps	True, unpredicted overlaps	Incorrectly predicted overlaps	Switching Error	
							Lower bound	Upper bound
Uniform	e^{Δ}	8048	8038	8005	43	33	1.07	47.7
	Δ		8145	7742	306	403	13.2	59.3
	1		8151	7643	405	508	18.3	62.5
Nonuniform	e^{Δ}	6338	6328	6288	50	40	1.78	47.2
	Δ		6387	5974	364	413	15.5	55.5
	1		6379	5829	509	550	20.1	54.2

Fig. 6. The number of real, predicted, correctly predicted, incorrectly predicted, and unpredicted overlaps for the two instances and three network flow methods (cost functions). Most Parsimonious Quasispecies Assembly is denoted by cost function 1 and Minimum Cost Quasispecies Assembly are denoted by cost Δ and e^{Δ}. The upper and lower bounds are given for the switching error, i.e. the average number of switches made by a true quasispecies when weaved through the predicted quasispecies.

The table on Figure 6 gives the total number of true overlaps and the total number of predicted overlaps. Then we give the number of true and false overlaps among predicted overlaps as well as the number of true overlaps which are missed by our method. Our experiments show that Most Parsimonious Assembly is the furthest from the true quasispecies and that the exponential cost is superior to the Δ-cost.

Following the error measures for diploid organism phasing we also estimate the *switching error* computed as follows. For each true quasispecies we identify the path in the transitively reduced read graph and count how many times it switches between predicted quasispecies. We then average number of switches over all true quasispecies. Since we output only the predicted read overlaps rather than quasispecies we estimate the switching error by providing the lower and upper bounds. The lower bound is given by the average number of unpredicted overlaps that occur along the path corresponding to a real quasispecies. The upper bound adds to the lower bound the average number of "forks" that occur along such a path, i.e. the number of reads that belong to two or more quasispecies simultaneously and therefore can potentially mislead us into switching. Our results show that the exponential cost is superior to Δ-cost as well as the most parsimonious solution and that our method admits only very small fraction of possible errors.

The table on Figure 7 gives the runtimes for the instance cleaning, transitive reduction, and linear programming subroutines in the program. As the table indicates, our method can deliver results in a reasonable amount of time, and is expected to scale well to the sizes of real problem instances.

References

1. Von Hahn, T., Yoon, J. C., Alter, H., Rice, C. M., Rehermann, B., Balfe, P., Mckeating, J. A. Hepatitis C Virus Continuously Escapes From Neutralizing Antibody and T-Cell Responses During Chronic Infection In Vivo, Gastroenterology 2007;132:667678.

Instance	Runtime (in seconds)		
	Cleaning	Transitive reduction	Linear Program
Uniform	29.18	71.36	80.91
		75.53	58.24
Nonuniform	14.9	35.8	51.56
		36.65	35.61

Fig. 7. The runtimes for major subroutines in the program. All runtimes were recorded on a modern machine with an Intel Core Duo 2 CPU and 2 GB of RAM.

2. Lippert, R., Schwartz, R., Lancia, G., and Istrail, S. (2002) Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem, *Briefings in Bioinformatics* **3(1):** 23–31.
3. Alekseyev, M.A., Pevzner, P.A. Colored de Bruijn graphs and the genome halving problem. *IEEE/ACM Trans Comput Biol Bioinform* **4(1)**:98-107.
4. Chaisson, M. J. and Pevzner, P. A. (2007) Short read fragment assembly of bacterial genomes, *Genome research* to appear.
5. Sundquist, A., Ronaghi, M., Tang, H., Pevzner, P., and Batzoglou, S. (2007) Whole-genome sequencing and assembly with high-throughput, short-read technologies, *PLoS ONE* **2(5)**:e484.
6. Brinza, D. and Zelikovsky, A. (2006) 2SNP: Scalable Phasing Based on 2-SNP Haplotypes, *Bioinformatics*, **22(3)**: 371-373.
7. 454 Lifescience (2007) *http://www.454.com/*.
8. Margulies M. et al. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437(7057)**:376-80
9. Albert, R., DasGupta, B., Dondi, R., Sontag, E., Zelikovsky, A. and Westbrooks, K., (2007) Signal Transduction Network Inference from Indirect Experimental Evidence, *Journal of Computational Biology*, **14(7)**:927-949.
10. Goldberg, A. (1997) An Effcient Implementation of a Scaling Minimum-Cost Flow Algorithm, *Journal of Algorithms* **22(1)**:1-29.
11. IG Systems CS2 Software (2007) *http://www.igsystems.com/cs2/*