Reconstructing cancer karyotypes from short read data: the half full and half empty glass

Rami Eitan and Ron Shamir[1]

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, 69978 Israel
rshamir@tau.ac.il, rami.eitan@gmail.com

June 12, 2017

Abstract

Background: During cancer progression genomes undergo point mutations as well as larger segmental changes. The latter include, among others, segmental deletions duplications, translocations and inversions. The result is a highly complex, patient-specific cancer karyotype. Using high-throughput technologies of deep sequencing and microarrays it is possible to interrogate a cancer genome and produce chromosomal copy number profiles and a list of breakpoints ("jumps") relative to the normal genome. This information is very detailed but local, and does not give the overall picture of the cancer genome. One of the basic challenges in cancer genome research is to use such information to infer the cancer karyotype.

We present here an algorithmic approach, based on graph theory and integer linear programming, that receives segmental copy number and breakpoint data as input and produces a cancer karyotype that is most concordant with them. We used simulations to evaluate the utility of our approach, and applied it to real data.

Results: By using a simulation model, we were able to estimate the correctness and robustness of the algorithm in a spectrum of scenarios. Under our base scenario, designed according to observations in real data, the algorithm correctly inferred 69% of the karyotypes. However, when using less stringent correctness metrics that account for incomplete and noisy data, 87% of the reconstructed karyotypes were correct. Furthermore, in scenarios where the data were very clean and complete, accuracy rose to 90%-100%. Some examples of analysis of real data, and the karyotypes reconstructed by our algorithm, are also presented.

Conclusion: While reconstruction of complete, perfect karyotype based on short read data is very hard, a large portion of the reconstruction will still be correct and can provide useful information.

Keywords: cancer, karyotypes, genome rearrangements, structural and numerical variations, deep sequencing, reconstruction, graph theory, integer linear programming.

[1] corresponding author

Background

The current understanding of cancer suggests that it is a disease driven by somatic mutations that accumulate in the genome, within a certain tissue, during the lifetime of an individual. These mutations vary in size and effect. They can be small, e.g., single nucleotide mutations, or large structural variations caused by rearrangements such as deletions, inversions, tandem duplications and chromosomal translocations, or duplication and losses of entire chromosomes [1]. Over time these rearrangements accumulate and result in genomes less and less similar to the germline genome.

Cancer genomes are often described in the form of karyotypes. A *karyotype* is a high level description of the genome as a set of chromosomes and the number of copies of each. Normal karyotypes have two copies of each chromosome 1 to 22 and the sex chromosomes. In contrast, in cancer karyotypes some chromosomes may contain fragments originating from several normal chromosomes.

Types of aberration events Most segmental changes that happen during the progression of the disease can be categorized as deletion, tandem duplication, inversion, translocation, and deletion and duplication of entire chromosomes.

A *deletion* is characterized by a missing segment of a chromosome, a *tandem duplication* happens when part of the chromosome is duplicated and thus two copies of a segment appear where normally there would only be one. An *inversion* occurs when a segment of a chromosome is reversed relative to its original orientation (Figure 0).

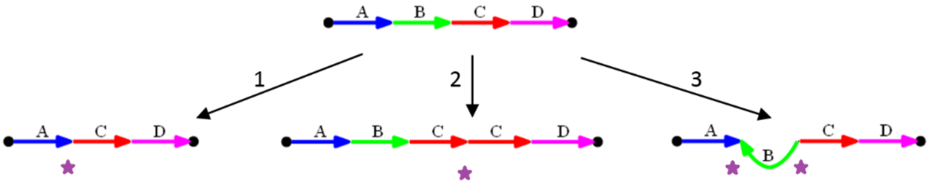

Figure 0: Basic types of rearrangements. (1) Deletion: segment B of the normal chromosome is deleted. (2) Tandem duplication: segment C duplicates and repeats. (3) Inversion: segment B is inverted. Stars indicate breakpoints.

A *translocation* happens when two different chromosomes "switch" end segments. Schematically, a translocation on two chromosomes (A,B) and (C,D) produces the chromosomes (A,D) and (C,B). A *whole chromosomal duplication (deletion)* adds (removes) a copy of a complete chromosome.

Breakpoints The molecular mechanisms that cause somatic genome rearrangements are still the focus of investigation. The main paradigm is that a genome rearrangement occurs when one or more chromosomes break and a following joining event reassembles the fragments in a different order. A *breakpoint* is defined as a genomic location where the normal DNA sequence is interrupted and two non-adjacent sequence segments appear consecutively due to a joining event. A breakpoint can be considered as the most basic unit of rearrangement. The stars in Figure 0 indicate breakpoints.

Models of genomic distance Modeling the somatic evolution of cancer holds great value for understanding the disease process. In 1995 Hannenhalli and Pevzner proposed a method to calculate the genomic distance between two species based on the minimal number of reversals (or reversals and translocations, in the multi-chromosome case) required to transform the genome of one species to another [2,3].

Braga et al. proposed another distance metric between genomes. They developed a method that calculates the distance between two genomes based on *Double-Cut and Join (DCJ)* operations and *indels* (Insertions and deletions), and utilized it to show evidence for deletion clusters in six species of Rickettsia [4]. Feijão et al. defined another metric based on *Single-cut and join (SCJ)* operations*,* and by using it they were able to recover between 60 and 90 percent of the topology of a phylogenic tree with 200 different genomes and with as many as 3000 genes [5,6]. Zeira and Shamir defined a generalized model, called *SCJD*, allowing the operations of cut, join and whole chromosome duplication. They developed a linear time algorithm for computing the shortest sequence of operations transforming one linear genome with one copy per gene into another with two copies per gene [7].

Ozery-Flato and Shamir introduced the *elementary distance* between two karyotypes, defined as the least number of elementary operations – breakage, fusion, duplication and deletion – transforming one into the other. They suggested a polynomial time 3-approximation algorithm to find the shortest elementary distance between two karyotypes. Applying the algorithm on some 58,000 karyotypes taken from the Mitelman database [8], 99.9% of the resulting solutions matched the lower (optimal) bound [9].

Detecting chromosomal aberrations

Paired end reads One of the main ways for inferring breakpoints in the genome, detecting structural variants and identifying rearrangements is using *paired end reads* produced by deep sequencing [10–13]. Paired end reads are generated by fragmenting the genomic DNA into short segments, followed by sequencing both ends of (some of) the segments (Figure 1). Typical lengths are ~350 bp per fragment (also called *insert*) and ~100 bp per read (*end*). The unsequenced segment of the insert is called the *gap* (length ~150 bp in the example above). The two ends of each read are then aligned back to a reference normal genome (in the case of cancer – the genome of a healthy cell from the same patient). The approximate length of the insert and the relative orientation of its ends is known in advance. We expect the two ends of a fragment to be aligned to the reference genome at roughly that distance and with the correct relative orientation. An alignment is called a *concordant* if it meets those conditions, and *discordant* otherwise.

Discordant reads suggest a breakpoint in the genome. A read taken from that spot will have its two ends aligned to locations on the reference genome where those positions originally lie. The type of discordance suggests the rearrangement event that occurred (See [14]).

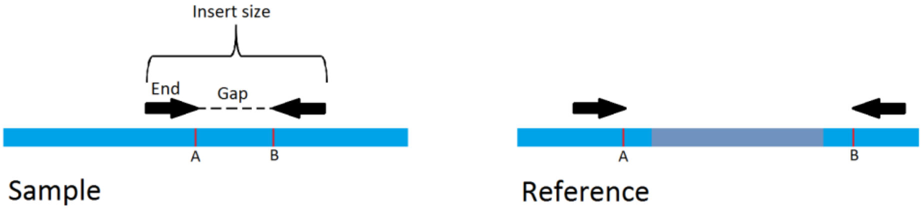

Figure 1: The paired ends read alignment signature of a deletion rearrangements. The grey area on the reference genome between points A and B was deleted in the sample genome. Any read whose gap falls between A and B on the sample genome will have its ends aligned to locations that are far apart on the same chromosome, indicating a deletion. Other rearrangements leave unique signatures in a similar manner.

Detecting structural variations A first step in analysis of paired end reads is their mapping to the reference genome. A variety of computational approaches were developed for inferring the structural variations from the discordant reads and produce a set of rearrangement events [15–20]. Other methods such as PREGO [21] take into account the concordant reads as well. BreaKmer [14] uses the misaligned reads together with the aligned concordant and discordant reads to predict rearrangements using k-mer statistics. CouGaR [22] is a method for identifying large-scale complex genomic rearrangements using both depth of coverage and discordant paired-ends mapping. SV-Bay [23] applies a Bayesian approach to data of mapped paired-ends reads to infer breakpoint locations and copy number variations and predict structural variations in a cancer genome. Recently, a new algorithm, Weaver [24], was proposed to estimate both the allelic copy number and inter-connectivity of SV's using a probabilistic graph model. Expanding on Weaver, Rajaraman et al. [25] used a graph model and an ILP formulation to further predict SV phasing and the interconnectivity of unphased SV's with high specificity. Other algorithmic approaches infer rearrangements that are less simple and have more complex signatures [14,26,27].

Some methods seek to achieve higher accuracy by aggregating results from several different tools. MetaSV [28] offers an improvement of accuracy and precision in detecting different kinds of structural variants. By effectively merging the results from multiple tools, they were able to reach F1-scores (harmonic mean of sensitivity and precision) of 96.2% for deletions and 84.7% for insertions. SomaticSeq [29] detects single nucleotide variants (SNVs) and small insertions and deletions (indels), using machine learning algorithms to incorporate the results from five somatic mutation callers. The authors report an F1 score of 90%.

Copy number variations

Duplications and deletions change the copy number (CN) of different segments of the DNA sequence, i.e. the number of times a segment is present in the karyotype. A normal (human) cell line has 22 diploid chromosomes (ignoring the sex chromosomes XX or XY) and so the CN of the entire karyotype is 2. A gain or a loss of an entire chromosome will decrease or increase the CN of that chromosome, respectively. A fraction of a chromosome can also be deleted or duplicated. The resulting segment or chromosome is said to have undergone a *copy number variation* (CNV).

Large CNVs can be detected by traditional methods like Fluorescence in-situ Hybridization (FISH) [30]. Higher resolution detection of CNVs can be achieved by Array Comparative Genomic Hybridization (aCGH) [31]. With the advent of next-generation sequencing (NGS), several methods have been developed to infer CNV's using DNA sequences [21,32,33]. NGS based methods have the potential to greatly increase the resolution of CNV analysis, but they present many computational challenges and different methods may still vary widely in the results they produce on the same DNA sequence [34].

Graph models for rearrangements

Graph theory has been highly instrumental in the area of genomic rearrangements. For example, de Bruijn graphs are used for genome assembly problems [35], and breakpoint graphs are used in reconstructing rearranged genomes across species [2,36]. More recently, similar methods were adapted for cancer genomes [9,37]. The breakpoint graph, introduced by Pevzner and Bafna in 1993 to represent the relation between two permutations of the same set of elements [38], remains today one of the key models in the study of genomic rearrangements.

Greenman et al. expanded on the breakpoint graph and introduced a construction that is essentially equivalent called the *allelic graph* and its counterpart the *somatic graph* [39].

Oesper et al. proposed a construction that expands on the breakpoint graph, called interval adjacency graph [21]. The interval adjacency graph is constructed directly from CN and breakpoint data. The discordant reads are used to infer breakpoint locations on the DNA sequence and partition it to intervals accordingly. A full description of the graph appears in Section 0.

Using the interval adjacency graph it is possible to infer rearranged sequences that agree with the data. Oesper et al. showed that an Eulerian path on the graph alternating between interval edges and reference / variant edges corresponds to a rearranged sequence of the chromosome. They developed an algorithm called *PREGO* to determine the most likely sequence of a rearranged karyotype. Using simulations they showed their algorithm can deduce the correct multiplicity of more than 80% of the variant edges, even with high noise and when the sample is heterogeneous. Furthermore, they applied PREGO to five ovarian cancer genomes and were able to identify numerous rearrangements and structural variants, some of which were consistent with known mechanisms. PREGO combines CN and adjacency information from paired end reads to infer multiplicity of different segments in the cancer genome. However except in simple cases, the underlying karyotype cannot be uniquely resolved, as many reconstructions will be consistent with the data.

Methods

We propose here a novel method that receives as input discordant paired-end reads and genomic CNs obtained from sequencing a cancer genome, and reconstructs a karyotype that is in most agreement with the input. The outline of our approach is as follows. We use the two data types together to construct a *bridge graph*, akin to the adjacency graph proposed by Oesper et al. [21]. An integer linear programming (ILP) optimization problem is formulated and then solved on the graph. The solution is a valid karyotype of the rearranged genome that is most concordant with the observed data. We also present the solution graphically.

The adjacency and bridge graphs

In our problem setup there is a *normal (or reference) genome*, whose contents is known, and an unknown *target genome* that should be reconstructed. A *breakpoint* is a point along the reference genome involved in a structural change event in the target genome.

Let C be the set of chromosomes in the reference karyotype. The breakpoints partition each chromosome $c \in C$ into a set of k^c intervals $I_c = \{I_1^c, I_2^c \dots I_{k^c}^c\}$, such that each $I_{k^c}^c$ is an interval between consecutive breakpoints, or between a breakpoint and a chromosome end. The intervals are numbered in increasing order along c, so that c is equal to the concatenation of the intervals $I_1^c, I_2^c \dots I_{k^c}^c$. We call the start and end points of interval I the *tail* and *head* of I and denote them by t_I and h_I respectively. Hence, $I = [t_I, h_I]$, and $-I = [h_I, t_I]$ is the interval I reversed. An *extremity* is a tail or a head of an interval. The set of all intervals $\mathcal{I} = \cup_{c \in C} I_j^c$ constitutes the set of the basic building blocks of the reference and target genomes. The length of interval I_j (in bases) is denoted by l_j, and $L = \sum l_i$ is the total length of all intervals.

The target genome can be represented by a set of chromosomes, where each chromosome is a sequence of intervals, some possibly reversed (Figure 3). A *bridge* is a pair of extremities that are not adjacent on the reference genome but are adjacent in the target genome. Bridges can be detected based on the paired-end read data of the target genome (Figure). The *support level* of bridge b_i is the number of paired-end reads that support it, denoted μ_i. The total support score for all bridges is denoted $\mu = \sum_{b_i} \mu_i$.

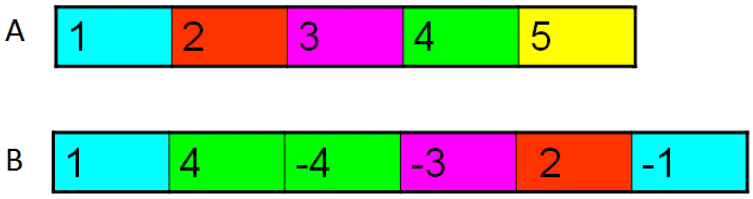

Figure 3: Reference and target genomes. A: reference (germline) chromosome segmented into intervals separated by breakpoints. B: The rearranged chromosome represented by the series of intervals 1,4,-4,-3,2,-1. Genome B contains the bridges $\{h_1, t_4\}, \{h_4, h_4\}, \{t_3, t_2\}$ and $\{h_2, h_1\}$. Note that $\{t_4, h_3\}$ is not a bridge.

Each interval $I_i \in I$ has a *CN* $N_i \geq 0$ indicating the number of times it appears in the target genome. The set of CNs of all intervals is called the *copy number profile* of the target. That profile can be derived from deep sequencing data or from array CGH data. In perfect data, N_i is exactly the number of copies of the interval in the target genome. In practice, the CNs are real valued estimates based on mean coverage of each interval.

Let us first reiterate the definition of the *interval adjacency graph*, introduced in [21]. The input is (1) the reference genome represented as a sequence of intervals for each chromosome. These intervals form the set $\mathcal{I} = \{I_1, \dots, I_n\}$; interval I_j has length l_j. (2) The CN profile of the intervals: Interval I_j has CN N_j. (3) The set of bridges $\{a_i, b_i\}_{i=1}^{m}$ and the support μ_i for each bridge. Each a_i and b_i is an extremity of an interval in $\mathcal{I}$. We define a weighted undirected graph $G(V, E, w)$ whose vertices are the interval extremities. For each interval $I_i = [t_i, h_i]$, the graph contains an *interval edge* $e_I(t_i, h_i) \in E_I$ connecting its two extremities, of weight N_i. For each two intervals I_i, I_{i+1} that are adjacent on the reference genome, a *reference edge* $e_R(h_i, t_{i+1}) \in E_R$ connects the head of I_i to the tail of I_{i+1}. Reference edges are unweighted. Each bridge is represented by a *bridge (or variant) edge* $e_V(a_i, b_j) \in E_V$ connecting the two extremities a_i and b_j, with weight μ_i. In total, the edge set of the graph is $E = E_I \cup E_R \cup E_V$. We denote by $S \subseteq V$ the set of vertices that represent *telomere nodes*, i.e. the nodes representing start and end points of each reference chromosome, hence $S = \cup_{c \in C} \{t_1^c, h_{k^c}^c\}$ includes the heads of all starting intervals and the tails of all ending intervals in each chromosome's partition.

A *bridge graph* is an interval adjacency graph with two minor changes: (1) bridge edges are assigned weights. The weight $w(e)$ of the bridge $e(u, v)$ is its *support score*, namely the number of paired end reads supporting that bridge. Hence, in a bridge graph both bridge and interval edges have weights. (2) We transform each undirected edge $e(u, v)$ in the interval adjacency graph into two directed edges $e_{\rightarrow}: u \rightarrow v, e_{\leftarrow}: v \rightarrow u$. The original undirected edge is referred to as a *connection* to distinguish it from the directed edges, and $E = E_{\rightarrow} \cup E_{\leftarrow}$ is the set of edges in the graph. An example of a bridge graph is given in Figure 4.

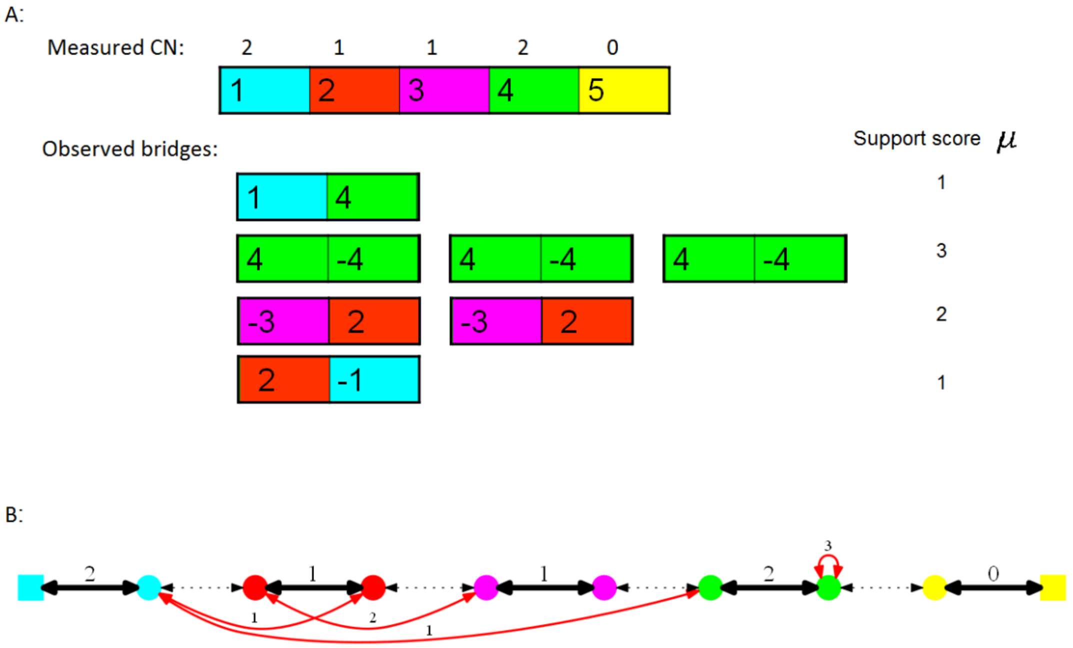

Figure 4: Bridge graph. The normal karyotype is the single chromosome $(1,2,3,4,5)$*. A: The measured CN and bridge data with the observed support score for each bridge. B: The corresponding bridge graph with weights for interval and bridge edges. All connections are composed of two antiparallel directed edges.*

Reconstructing the rearranged karyotype

Given the bridge graph $G(V,E,w)$, we wish to find paths in G that correspond to rearranged chromosomes. Suppose first that the input data are complete and errorless. Recall that $S \subseteq V$ is the set of vertices that represent *telomere nodes*, i.e. the nodes representing the start and end points of each chromosome. A *valid path* p is a path through G beginning and ending at $s_1, s_2 \in S$ that alternately traverses interval and non-interval (i.e. reference/bridge) edges, and where the number of times each interval connection e_i is traversed (in either direction), denoted $f_p(e_i)$, is less than or equal to the CN of interval i, N_i.

The requirement for an alternating path is because a traversal of an interval edge corresponds to traversing a segment from the reference genome, while a traversal of a reference/bridge edge is equivalent to a transition between segments. Therefore, such an alternating path represents a sequence of segments from the reference genome. Note that $f_p(e_i) = f_p(e_{i\rightarrow}) + f_p(e_{i\leftarrow})$ for every connection e. A set of such paths $P = \{p_1, p_2 \dots p_n\}$ where for each interval connection e_i, $\sum_{p\in P} f_p(e_i) = N_i$ corresponds to a set of rearranged chromosomes, or a valid karyotype.

The restriction that the path alternates between interval and non-interval edges means that at each non-telomeric node $v \notin S$, every traversal on an interval edge going into v must be followed by a traversal on a reference\bridge edge going out of v, and vice-versa. Telomeric nodes are excluded from this constraint as by definition they are the start or end of a path.

As detailed above, each connection between nodes u, v is composed of two antiparallel directed edges. For each node $v \in V$ we denote $E_{I_\leftarrow}(v), E_{I\rightarrow}(v), E_{R\leftarrow}(v)$, $E_{R\rightarrow}(v), E_{B\leftarrow}(v), E_{B\rightarrow}(v)$ as the set of interval, reference and bridge edges that go in and out of v respectively. As above, we denote by $f_p(e)$ the number of times a connection e is traversed in path p and $f_P(e) = \sum_{p\in P} f_p(e)$ is the total number of times a connection e is traversed in P. Additionally, for a set of connections E, $f_P(E) = \sum_{e\in E} f_P(e)$ is the total number of times all connections in E are traversed in P. The constraints for a valid set of paths P, representing a rearranged karyotype, can be therefore formulated as:

(1) $f_P\left(E_{I_\rightarrow}(v)\right) = f_P\left(E_{R_\leftarrow}(v)\right) + f_P\left(E_{V_\leftarrow}(v)\right)$
$\forall_{v \notin S}$

(2) $f_P\left(E_{I_\leftarrow}(v)\right) = f_P\left(E_{R_\rightarrow}(v)\right) + f_P\left(E_{V_\rightarrow}(v)\right)$
$\forall_{v \notin S}$

(3) $f_P(e) \in \mathbb{N}^0$
$\forall_{e \in E}$

Scoring candidate solutions Recall that the interval and bridge edges have weights, representing the measured CN of the intervals and the support score for the bridges, respectively. These values are in practice noisy. Given a bridge graph $G(V,E,w)$ and a valid set of paths P representing a rearranged karyotype, we define a *discordance score* of P, denoted $d_G(P)$, which measures how much P is in agreement with the data in G, as follows:

$$d_G(P) = \sum_{e \in E_I} \frac{l_e}{L} |f_P(e) - w(e)| + \alpha \sum_{\substack{e \in E_V \\ e \notin P}} \frac{w(e)}{\mu}$$

The first sum measures the disagreement of P with the CN profile. It is the sum over all interval edges $e \in E_I$ of the absolute difference between $f_P(e)$ and the input weight $w(e)$, normalized by l_e. We normalize the weights of the intervals by their lengths since longer genomic intervals are expected to have more accurate CN values, and hence should be penalized more for disagreement. Dividing by L guarantees that the range of the first sum is $[0,1]$ if the absolute difference values are ≤ 1.

The second sum the disagreement of P with the bridge data. The more bridges P is utilizing, the more concordant it is with the bridge data. To reflect this, a penalty is given for each bridge edge $e \in E_V$ that is not used in P. The bigger the support score for a bridge is, the bigger the penalty if it is not used, and so the penalties are normalized by $w(e)$. Dividing by $\mu = \sum_{e \in E_V} w(e)$ guarantees that the range of the second sum is $[0,1]$. To avoid summing over $e \notin P$, we can rewrite the second term as $\alpha \sum_{\mathrm{e} \in \mathrm{E_V}} \frac{\mathrm{w(e)}}{\mu}\left(1 - \min\left(1, \mathrm{f_P(e)}\right)\right)$.

The parameter α determines the relative weight the algorithm gives to paired-end reads data, i.e. how much it tries to utilize bridge edges in the solution. Using the algorithm on real tumor data, we set $\alpha = 0.5$.

The ILP formulation We wish to find a rearranged karyotype that is most consistent with the data, i.e., it corresponds to a valid set of paths and has smallest possible discordance score. This problem can be formulated as an ILP on the bridge graph $G(V,E,w)$, as we now show.

For each connection $e_i \in E$ we define two variables $x_{i\rightarrow}, x_{i\leftarrow}$. The variables represent the number of times each edge is traversed in a path, and so $f_P(e_i) = x_{i\rightarrow} + x_{i\leftarrow}$. Each variable is noted x^I, x^B or x^R for interval, bridge or reference edges respectively. Using these variables we can formulate the problem as follows:

Minimize:

$$d_G(f_P) = \sum_{e\in E_I} \frac{l_e}{L} |x^I_{e\rightarrow} + x^I_{e\leftarrow} - w(e)| + \alpha \sum_{e\in E_V} \frac{w(e)}{\mu}\left(1 - min(1, x^B_{e\rightarrow} + x^B_{\leftarrow})\right)$$

Subject to:

(1) $\forall_i x_i \in \mathbb{N}^0$

(2) $\forall_{v\notin S}\ \sum_{e_i\in E_{I_\rightarrow}(v)} x^I_{i\rightarrow} = \sum_{e_i\in E_{R_\leftarrow}(v)} x^R_{i\leftarrow} + \sum_{e_i\in E_{V\leftarrow}(v)} x^B_{i\leftarrow}$

(3) $\forall_{v\notin S} \sum_{e_i\in E_{I_\leftarrow}(v)} x^I_{i\leftarrow} = \sum_{e_i\in E_{R_\rightarrow}(v)} x^R_{i\rightarrow} + \sum_{e_i\in E_{V\rightarrow}(v)} x^B_{i\rightarrow}$

Constraint set (1) guarantees an integral non-negative solution. Constraints (2) and (3) are the valid path constraints. Note that telomeric nodes in S are not constrained.

Tools

The core of the algorithm was implemented in java using the ILP solver package CPLEX, distributed by IBM [40] and was run on UNIX. The simulations module and the rest of the algorithm was implemented in python version 2.7 on Windows. The code is available in https://github.com/Shamir-Lab/Karyotype-reconstruction. A typical run of a single karyotype on a standard PC takes around 1 second.

Results and discussion

Simulations

To assess the performance of our algorithm, we simulated tumor karyotypes and applied the algorithm to them. To evaluate the quality of each reconstructed karyotype, it was compared to the correct karyotype, and summary statistics were computed. An overview of the simulation algorithm is as follows:

1. Start with a normal diploid karyotype H with C chromosomes
2. Perform K operations resulting in karyotype T'
3. Compute the exact (noiseless) CN profile and the bridges in T'
4. Add noise to the CN data and generate support values for the bridges

We start with a normal diploid karyotype H with a prescribed number of chromosomes. For simplicity, each chromosome is represented by a sequence of 300 atomic segments, which are its basic units. We perform a series of operations on the karyotype by applying deletions, inversions, tandem duplications and translocations. The types and the positions of the rearrangements are drawn uniformly at random. The span of operations that affect a single chromosomes (deletions, duplications and inversions) was limited to 30 atomic segments. This limit was set in order to avoid rapid erasure of large chromosomal segments by deletions. The total number of operations applied varies and determines the complexity of the resulting tumor karyotype T.

By comparing H and T, breakpoints are detected and each normal chromosome is partitioned into segments. Each segment has a CN (the number of occurrences of that segment in T). Each two consecutive segments in T that are not consecutive (and/or not in the same relative orientation) in H constitute a bridge. The clean (noiseless) data can thus be summarized as an integer-valued CN profile and the set of all bridges formed.

To simulate noisy scenarios, the CN profile and the bridge information is modified as follows. Normally distributed noise x is added to the CN of each segment independently, where $x \sim N(0, \epsilon)$. The support for each bridge (corresponding to the number of discordant reads supporting it) is drawn independently from an exponential distribution $Exp(\lambda)$ (The exponential distribution was chosen based on empirical data with $\lambda = 0.1866$. See below). To simulate the possibility of bridges being completely missed, each bridge has probability p to completely be omitted from the final set of bridges.

In summary, the simulation program receives the following parameters (the default values appear in parentheses):

- C - The number of chromosomes (default: 5).
- N - The number of structural and numerical operations applied (default: 5).
- ϵ - The standard deviation of the noise in the CN profile data (default: 0.28)
- p – The probability to completely miss a bridge (default: 0.05).

In the *base scenario*, all parameters were at their default values. These parameters correspond to those computed on a tumor sample of medium complexity and a realistic level of noise (see Real tumor analysis below). Other scenarios were explored by changing the value of one of the parameters above while keeping the rest at their default levels.

Solution quality measures We used five different measures for the level of correctness of a solution. Let T be the simulated (true) karyotype, let T^* be the simulated noisy karyotype, and let S be the karyotype produced by the algorithm:

1. Is S equivalent to T? We say that S is *equivalent* to T if they have the same CN profile and both use the same bridges. Most equivalent karyotypes only differ in chromosomal orientation, and thus represent the same solution. We call such a solution *correct.*
2. Do S and T have the same CN profile? The CN of an interval is determined by many reads (or probes) and so is expected to be more robust than bridge information, determined by a few paired end reads. This criterion tests if S and T match in their CN profile. We call this criterion *Equal Copy Number* (ECN).
3. Does S have an equal or better score than T? When noise level is high, T and T^* may differ substantially, and a solution closer to T^* than to T does not indicate a failure of the algorithm but rather that the noise level is too high. Here the score is the ILP objective function value. We call this criterion *Equal or Better Score* (EBS).
4. Is S equivalent to T excluding missing bridges? T^* may not include all the bridges found in T, and in that case S can never be equivalent to T. However, we consider S to be correct for all observed bridges if it has the correct CN profile for all segments that are unaffected by a missed bridge, and is using all the bridges from T that are included in T^*(Figure S2). We call this metric *Equivalent for Observed Bridges* (EOB).
5. What fraction of the intervals has the correct CN? This score is the percentage of intervals, weighted by length, that have the same CN in S and T. Unlike criteria 1-4, which are binary, this criterion measures the extent of correctness of a solution, and thus is more sensitive and accounts also for partially-correct solutions. We call it the *CN score.*

Base scenario 10,000 karyotypes were generated for the base scenario, and the algorithm was applied with bridge support weight $\alpha = 0.1$. The performance is summarized in Figure .

To assess the distribution of each success rate criterion, the karyotypes were divided into 100 batches of 100 karyotypes each. Mean scores were captured for each batch and the variation of the mean was computed.

The algorithm correctly identified between 55% and 73% of the karyotypes in each batch, with an average of 62%. For an additional 13% of the cases, the solution had an equal CN profile as the correct solution, reaching a total of 75%. An average of 82% of all karyotypes resulted a solution with a score equal or better than the correct one. When disregarding missing bridges, the algorithm correctly identified an average 84% of karyotypes. The mean CN score of all the 10,000 simulations was 0.97 with a small standard deviation of 0.009.

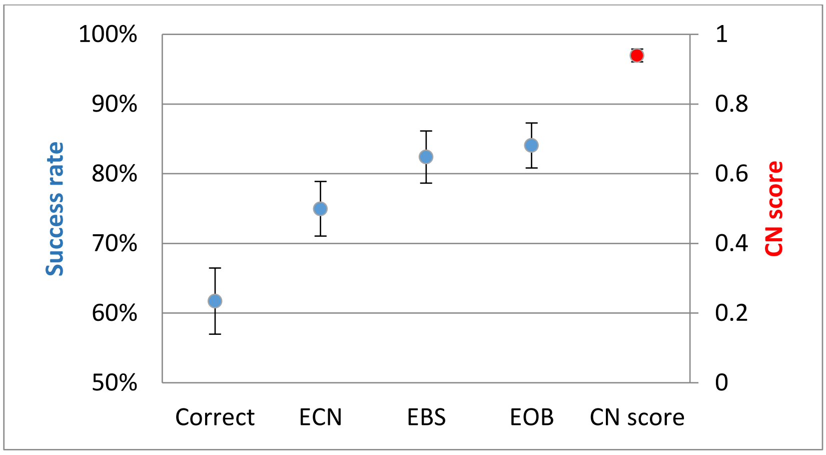

Figure 5: Distribution of the success rate over 100 independant simulations of the base scenario. Error bars are ± the standard deviation.

The effect of separate parameters The effect of separate parameters was tested by simulations in which one parameter was altered, while keeping the other parameters at their value in the base scenario. 100 simulated karyotypes were generated for each value and the percentage of solutions falling into the categories of correct, ECN, EBS and EOB was evaluated.

Bridge support weight We first tested the effect of α, the relative weight assigned the bridges, on the performance, for $0 \leq \alpha \leq 2$. There is a noticeable improvement when $\alpha > 0$, and little effect for the range of $0 < \alpha \leq 0.1$. For larger values of α there is a small but noticeable negative effect. (Table S1).

Noise in copy number measurements We tested the algorithm for different levels of CN noise ϵ under the base scenario. The results are shown in Figure 6. As expected, a higher level of noise makes it harder for the algorithm to find the correct solution. For $\epsilon \leq 0.3$ the performance of the algorithm is quite good, and for $\epsilon \geq 0.4$ the results begin to deteriorate. As expected, at high noise levels the majority of the solutions have better score than the true one.

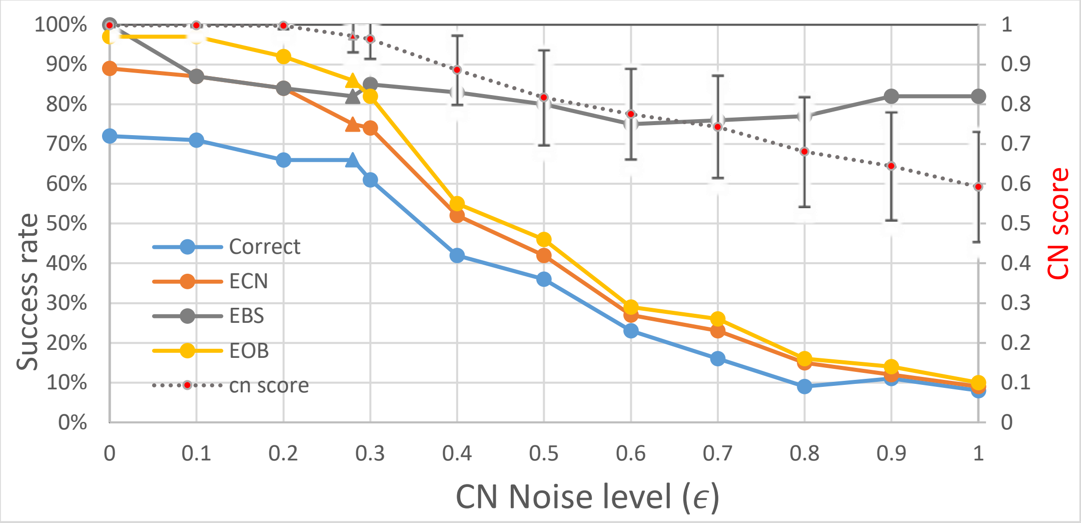

Figure 6: Performance of the algorithm as a function of noise level. For the CN score, the bars represent ± 0.5 *std. Data points for the default value of* $\epsilon = 0.28$ *are marked with a triangle.*

The number of operations We tested the algorithm on karyotypes that underwent $1 \leq N \leq 30$ structural and numerical operations, under the base scenario. The results are shown in Figure 0. As expected, more operations make the problem harder and the success rates decrease. For example, the fraction of perfectly solved cases drops from 88% with one operation to less than 10% with 30 operations. The CN score drops more slowly, as CN of long fragments can still be reasonably inferred even if their order is incorrect.

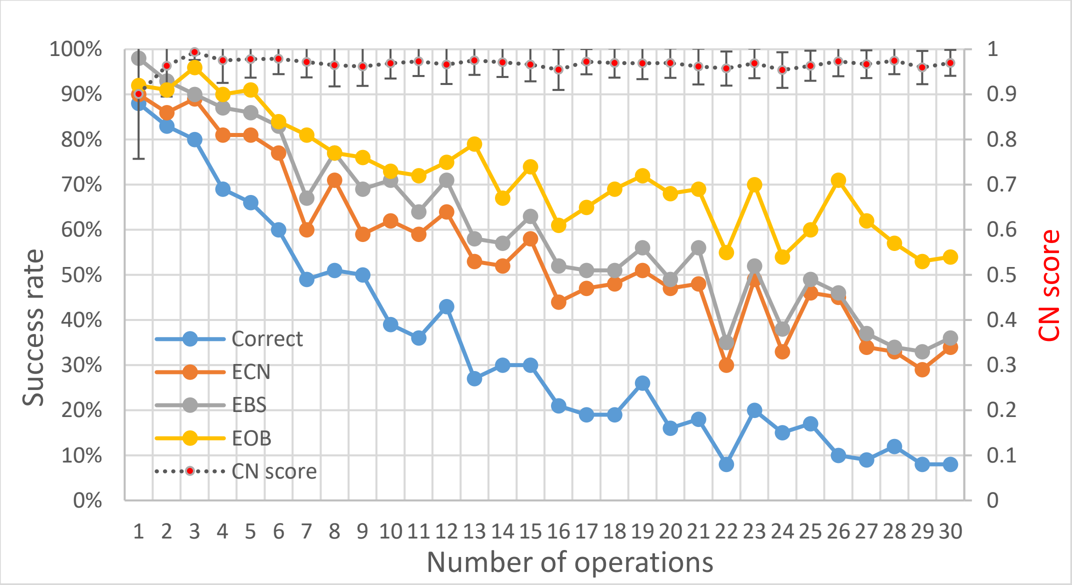

Figure 0: The effect of the number of operations. Success rates and CN scores. Error bars represent ± 0.5 *std.*

Other parameters When testing the effect of other parameters, the results met our expectations – karyotypes with less chromosomes (Figure S3) or a single copy of each chromosome instead of diploid (Figure S4) yielded better results. Results were also better when the probability of missing a bridge was lower (Figure S5).

We also looked at cancer heterogeneity situations. Different cells of the same tumor can have different karyotypes, having taken different evolutionary paths [41–45]. Most cancer data today is still based on DNA from numerous cells, providing measurements from a mixture of genomes. Can the karyotype be reconstructed out of the heterogeneous mixture? When simulating data mixture of normal and a cancer karyotype results only dropped mildly with the relative abundance of normal data (Figure S8). However, when mixing two distinct cancer karyotypes, performance dropped rapidly with the heterogeneity (Figure S6).

Finally, we simulated karyotypes by selecting operations with frequencies as reported in [46] rounded to multiples of 10%. There was little difference in the success rates between the uniform distribution and the uneven one (Figure S7).

Real tumor analysis

We applied the algorithm on data extracted from real samples. Malhotra et al. [46] examined whole genome sequencing data of 64 different tumor samples, and reported for each sample a CN profile and a set of bridges with their support. We first filtered from the data very small segments and the corresponding breakpoints (see Supplement). Often the set of normal chromosomes that are involved in rearrangements and CN changes in a tumor can be partitioned into independent groups of chromosomes (i.e., no two segments in different groups are connected by a bridge). In our graph representation, each such group is a connected component, which can be analyzed separately by the algorithm. The 64 tumor samples in [46] constituted together 570 such components, and each was analyzed separately.

Noise estimation We first wanted to assess the noise level in the actual data affecting the reported CN values. Since CN in noiseless data should be integer, we estimated the noise d_i for the reported CN c_i as $c_i - [c_i]$, where $[x]$ is the nearest integer value to x. The CN data include 22,321 CN segments. A scatter plot of the standard deviation of the noise level vs. the number of bridges in each component can be seen in Figure 8. As expected, the mean noise level across the data was 0, showing that the noise is unbiased towards neither negative nor positive values. The standard deviation was 0.28, a value that we used as our default simulation scenario. Note that this estimate is a lower bound, since some measured CN values may actually differ from the real ones by more than 0.5.

In addition to CNs, the data include bridges and for each bridge an integer value, its support. The expected average support can be derived from the read depth and the insert size (see Supplement) and was found to be 10.7. The observed mean support score across all the data was 10.8. Figure 9 shows the distribution of the support scores across the data. The distribution closely resembles an exponential distribution with $\lambda = 0.1866$. For that reason, that was the value used in our simulation model (see supplement for more details).

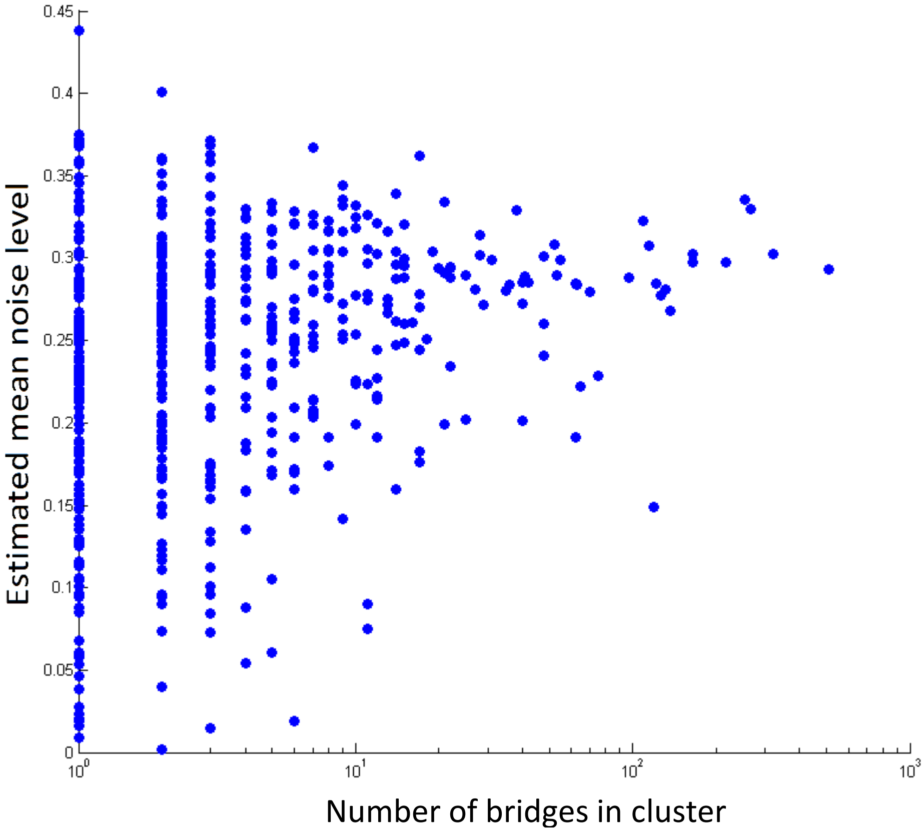

Figure 8: Estimated noise level in real cancer samples. The plot shows for each of the 670 components in the tumor samples in [46]*, the number of bridges and an estimate of the noise level calculated as standard deviation of the distances of the CN in the sample from the closest integer value.*

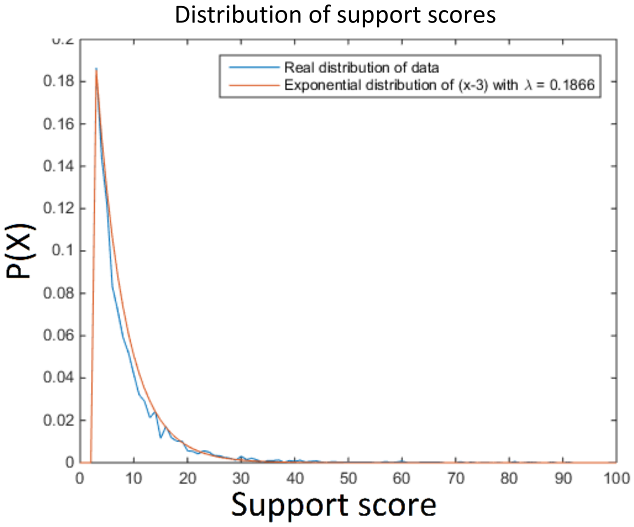

Figure 9: The distribution of the support score across the data plotted against an exponential distribution with $\lambda = 0.1866$*. In both distributions values below 3 are ignored.*

The GBM10 sample. We analyzed in detail three components of bridge graphs obtained from real data. Table S2 shows information about them. Each has undergone 7-8 rearrangements, involving 1-4 chromosomes. For each component, the ILP algorithm outputs a directed weighted graph with a weight function that minimizes the distance and that can be broken into a set of paths $P = \{p_1, \dots p_n\}$, starting and ending at a telomere nodes, and alternating interval and non-interval edges. Another script translates the solution of the ILP solver to a dot language representation [47] that can then be visualized using a graph visualization tool such as GraphViz [48].

Figure 10A shows the graph corresponding to the component of chromosomes 4 and X in tumor sample GBM 10 (Glioblastoma multiforme). The resulting karyotype produced by our algorithm for this example is shown in Figure 10B. This graph can be broken into four different paths, representing both copies of the rearranged chromosomes 4 and X (Figure 10C).

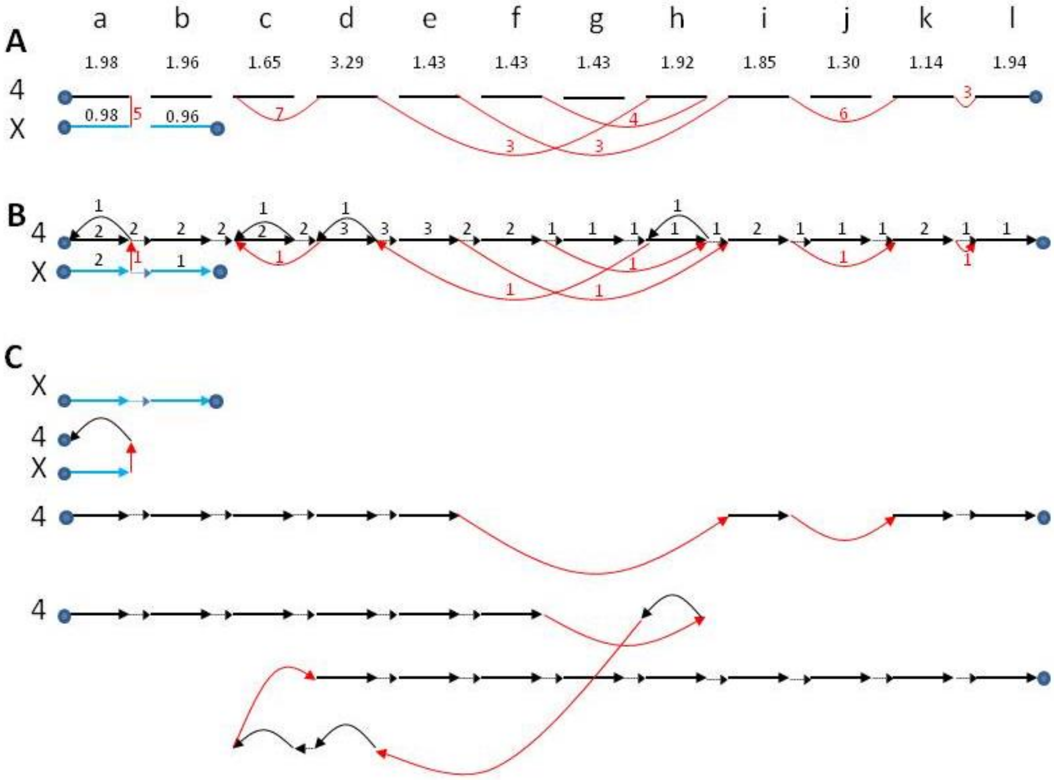

Figure 10: Results on sample GBM 10. The chromosomes were divided into segments according to the breakpoints inferred from the paired ends reads data and were named a-l. Segment sizes are not shown to scale. We mark interval, reference and bridge edges by black, dotted and red arcs respectively. The number next to a red edge (bridge) is the number of observed supporting reads for that bridge. In all subfigures the same intervals (here: a through l for Chr. 4 and a, b for Chr. X) are aligned. The numbers in the second line are observed coverage values. (A) Bridge graph for chromosomes X and 4. The bridge btween segments k and l is a result of breakpoint filtering (see Supplement). (B) Solution suggested by our algorithm. For this sample the average distance of the resulting karyotype from the data, weighted by segment length, is 0.28. Note that segments a, c, d, and h have edges in both directions suggesting the solution includes traversal of these segments in both directions. (C) The different paths comprising the solution, representing the rearranged karyotype of chromosomes 4 and X.

The other two examples are described in the Supplement.

Conclusions

In this work, the problem of inferring a tumor karyotypes from short paired end read data was investigated. A novel algorithm based on graph theory and ILP was introduced to solve the problem, and simulations were performed in order to evaluate the utility of such an approach. Some examples of analysis of real data were also presented.

To accurately estimate the correctness and robustness of the algorithm, validation against a data set of verified karyotypes is needed. However, a comprehensive set of sequenced tumor samples with CN profiles and paired-end reads data, matched with the corresponding true karyotypes, is currently not available. Data sets that currently exist either do not include a fully reconstructed karyotype, or include karyotypes of a very low resolution [8]. We therefore used simulations to test and measure the success of our algorithm in a spectrum of scenarios, as well as to point out potential pitfalls.

The analysis of simulated data suggests that the most meaningful factors affecting the accuracy of solutions produced by our method are the noise and completeness levels of the data. We tested the algorithm in a scenario, designed to mimic parameter values observed in real data. Under these conditions, the algorithm correctly inferred 69% of the karyotypes. However, the success rate increased to 79% when considering solutions that are correct relative to the noisy input, and when accounting for unreported bridges, 87% of the tested cases were correct (Figure 5).

Furthermore, in scenarios where there is almost no noise, or when no bridges are unreported, the results are much better: accuracy was 90% and 100%, respectively (Figure 6, Figure S5). This strongly suggests that our method is limited mostly by the completeness and accuracy of the measured data. It suggests that more accurate sequencing technologies are needed in order to increase the chance to solve the karyotype reconstruction problem correctly.

Our method was relatively robust when applied on data taken from tumor cells contaminated by healthy tissue (Figure S8). A sample that includes reads taken from a mixture of different tumor cells poses a bigger challenge, and the resulting karyotype is incorrect more often than it is correct (Figure S6).

Depending on one's perspective, the results can be viewed as good or bad news. On the one hand, full, perfect reconstruction is not attained in over 30% of the cases. On the other hand, even in those imperfect cases, most of the reconstruction details are correct, as quantified by the other, less stringent, measurement criteria (Figure 5). Biological research has a great tradition of building up from incomplete data, the most obvious example being the human genome whose yet-incomplete versions have kept evolving for the past fifteen years. It may be the case that the imperfect reconstruction of cancer karyotypes can still produce valuable conclusions and findings.

Limitations Using simulations allows us to gain better understanding of the capabilities and limitations of our algorithm, but it requires us to make assumptions about the mechanisms driving genomic rearrangements in tumor cells and about the statistical properties of the read data. Both types of assumptions limit the generality of conclusions we can draw.

Firstly, our model defines a limited set of possible rearrangements (deletion, duplication, inversion and chromosomal translocation) and assumes that they occur with equal probabilities. Furthermore, our simulation of rearrangement events (except translocations) limits the genomic range they can span (see section 0) and assumes that events are equally likely to occur in any position on the genome. While these assumptions are very far from the real process of mutating cancer cells, they do provide a mechanism that can generate any rearranged karyotype. Our method proved robust when adjusting the frequency of each type of rearrangement to that observed in the data obtained from [46] (Figure S7), but other possible rearrangement mechanisms and their effect on the performance of the algorithm were not explored.

A second problem arises when attempting to create very complicated karyotypes using a large number of rearrangements. Stephens et al [49] suggested that in some cases a single catastrophic event called *chromothripsis* occurs, in which a section of the chromosome is shattered into a large number of small fragments and then reassembled, creating a karyotype that is much more complex [49]. While all possible karyotypes can be generated using our model, very complex ones are unlikely. Note that once a deletion operation has been performed, the deleted segment cannot reappear and will therefore be absent from the final karyotype. When performing a large number of rearrangements on a chromosome, deletions will occur and sometime remove segments that were rearranged by a previous operation, essentially reducing the complexity of the resulting final karyotype. We tested our method on karyotypes that have undergone a maximum of 30 operations (Figure 0), but a modified simulation model needs to be used in order to generate more complex karyotypes. Currently our results reflect more faithfully the ability of the algorithm on relatively simple karyotypes, which constituted the majority in the real data that we used.

A third type of limitation is due to the noise model assumptions. While we tried to borrow values of noise as estimated from the real data (see Real tumor analysis), there are other parameters that affect the noise and thus the quality of the analysis, including incorrectly mapped reads due to sequencing errors, non-uniquely mappable reads, insert length variance, breakpoints that fall within a read (and not in the gap), non-uniform read coverage, etc. These are all left to future work.

One of the limitations of our algorithm is its inability to “predict” bridges that were not observed in the data. The algorithm looks for a path on the graph corresponding to a karyotype that best fits the observed CN profile, yet it overlooks potential paths that can be constructed by bridging two unconnected interval edges – essentially predicting a bridge. This implies that data produced using sensitive methods, even with higher rates of false positives, might be preferable over data with false negatives.

Future directions One important aspect of the technology in detecting bridges is the insert size. A bridge will usually be detected only when the two reads of a PER are on the two different sides of it (see supplement). Therefore, the larger the read length and insert - the higher the bridge coverage. This implies that sequencing techniques with longer inserts can dramatically change the performance of the algorithm. Several such techniques are forthcoming, and some methods for detecting structural variations were already developed for them [28] [29] [22]. Note however that very short rearrangements that span less base pairs than the length of the read may be missed altogether.

A possible extension to our method can be the addition of weights to the reference edges. Recall that reference edges represent a connection between two segments that is expected according to the reference genome. Unlike interval edges or bridge edges, reference edges are weightless in our model. One metric that can be used to establish a confidence score for a reference edge is the number of PERs whose ends span the two segments bordering the reference connection.

Declarations

Abbreviations

ILP: Integer Linear Programming; CN: Copy Number; CNV: Copy Number Variations; PER: Paired-End Reads; ECN: Equal Copy Number; EBS: Equal or Better Score; EOB: Equal for Observed Bridges.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and material

All Data analyzed in this research was reported by Malhorta et al. in [46] and is available for download as supplemental material in http://genome.cshlp.org/content/23/5/762/suppl/DC1

Competing interests

The authors declare that they have no competing interests.

Funding

This study was supported in part by the Israel Science Foundation (grant 317/13) and by the Israel Cancer Association.

Authors' contributions

RE and RS designed the study. RE prepared the data, developed the tools used to simulate and analyze data, and produced the results. RE and RS analyzed the results. RE and RS wrote the manuscript. Both authors read and approved the final manuscript.

Acknowledgments

We would like to thank Nir Atias for help with the IBM CPLEX package and Roy Kasher for help with software design and optimization.

References

1. Vogelstein, B. *et al.* Cancer Genome Lanscapes. *Science (80-.).* **339,** 1546–1558 (2013).

2. Hannenhalli, S. & Pevzner, P. a. Transforming men into mice (polynomial algorithm for genomicdistance problem). *Proc. IEEE 36th Annu. Found. Comput. Sci.* (1995). doi:10.1109/SFCS.1995.492588

3. Sridhar Hannenhalli, P. P. Transforming Cabbage into Turnip (polynomial algorithm for sorting signed permutations by reversals). *JACM* **46,** 1–27 (1999).

4. Braga, M. D. V, Willing, E. & Stoye, J. Double cut and join with insertions and deletions. *J. Comput. Biol.* **18,** 1167–1184 (2011).

5. Feijão, P. & Meidanis, J. SCJ: A breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **8,** 1318–1329 (2011).

6. Biller, P., Feijão, P. & Meidanis, J. Rearrangement-based phylogeny using the single-cut-or-join operation. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **10,** 122–134 (2013).

7. Zeira, R. & Shamir, R. in 396–409 (Springer, Cham, 2015). doi:10.1007/978-3-319-19929-0_34

8. Mitelman F, Johansson, B. & Mertens F (Eds.). Mitelman Database of Chromosome Aberrations and Gene Fusions in Cancer. (2016). at <http://cgap.nci.nih.gov/Chromosomes/Mitelman>

9. Ozery-Flato, M. & Shamir, R. Sorting cancer karyotypes by elementary operations. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **5267 LNBI,** 211–225 (2008).

10. Tuzun, E. *et al.* Fine-scale structural variation of the human genome. *Nat. Genet.* **37,** 727–32 (2005).

11. Korbel, J. O. *et al.* Paired-end mapping reveals extensive structural variation in the human genome. *Science* **318,** 420–6 (2007).

12. Kidd, J. M. *et al.* Mapping and sequencing of structural variation from eight human genomes. *Nature* **453,** 56–64 (2008).

13. Bentley, D. R. *et al.* Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456,** 53–9 (2008).

14. Abo, R. P. *et al.* BreaKmer: detection of structural variation in targeted massively parallel sequencing data using kmers. *Nucleic Acids Res.* **43,** 1–13 (2014).

15. Quinlan, A. R. *et al.* Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Res.* **20,** 623–635 (2010).

16. Chen, K. *et al.* BreakDancer: An algorithm for high resolution mapping of genomic structural variation. **6,** 677–681 (2013).

17. Korbel, J. O. *et al.* PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.* **10,** R23 (2009).

18. Hormozdiari, F., Alkan, C., Eichler, E. E. & Sahinalp, S. C. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res.* **19,** 1270–8 (2009).

19. Hormozdiari, F., Hajirasouliha, I., McPherson, A., Eichler, E. E. & Sahinalp, S. C. Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res.* **21,** 2203–12 (2011).

20. Hormozdiari, F. *et al.* Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics* **26,** i350-7 (2010).

21. Oesper, L., Ritz, A., Aerni, S. J., Drebin, R. & Raphael, B. J. Reconstructing cancer genomes from paired-end sequencing data. *BMC Bioinformatics* **13 Suppl 6,** S10 (2012).

22. Dzamba, M. *et al.* Identification of complex genomic rearrangements in cancers using CouGaR. *Genome Res.* **27,** 107–117 (2017).

23. Iakovishina, D., Janoueix-Lerosey, I., Barillot, E., Regnier, M. & Boeva, V. SV-Bay: structural variant detection in cancer genomes using a Bayesian approach with correction for GC-content and read mappability. *Bioinformatics* **32,** 984–992 (2016).

24. Li, Y., Zhou, S., Schwartz, D. C. & Ma, J. Allele-Specific Quantification of Structural Variations in Cancer Genomes. *Cell Syst.* **3,** 21–34 (2016).

25. Rajaraman, A. & Ma, J. in 224–240 (Springer, Cham, 2017). doi:10.1007/978-3-319-56970-3_14

26. McPherson, a. *et al.* nFuse: Discovery of complex genomic rearrangements in cancer using high-throughput sequencing. *Genome Res.* **22,** 2250–2261 (2012).

27. Baca, S. C. *et al.* Punctuated evolution of prostate cancer genomes. *Cell* **153,** 666–677 (2013).

28. Mohiyuddin, M. *et al.* MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. *Bioinformatics* **31,** 2741–4 (2015).

29. Fang, L. T. *et al.* An ensemble approach to accurately detect somatic mutations using SomaticSeq. *Genome Biol.* **16,** 197 (2015).

30. Kallioniemi, A., Visakorpi, T., Karhu, R., Pinkel, D. & Kallioniemi, O. Gene Copy Number Analysis by Fluorescence in Situ Hybridization and Comparative Genomic Hybridization. *Methods* **9,** 113–21 (1996).

31. Pinkel, D. & Albertson, D. G. Array comparative genomic hybridization and its applications in cancer. *Nat. Genet.* **37 Suppl,** S11-7 (2005).

32. Yoon, S., Xuan, Z., Makarov, V., Ye, K. & Sebat, J. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.* **19,** 1586–92 (2009).

33. Abyzov, A., Urban, A. E., Snyder, M. & Gerstein, M. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.* **21,** 974–84 (2011).

34. Alkan, C., Coe, B. P. & Eichler, E. E. Genome structural variation discovery and genotyping. *Nat. Rev. Genet.* **12,** 363–376 (2011).

35. Pevzner, P. a, Tang, H. & Waterman, M. S. An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. U. S. A.* **98,** 9748–9753 (2001).

36. Pevzner, P. *Computational Molecular Biology: An Algorithmic Approach*. (MIT Press, 2000). at <http://books.google.com/books?hl=en&lr=&id=dpgh2UxGpacC&pgis=1>

37. Raphael, B. J., Volik, S., Collins, C. & Pevzner, P. a. Reconstructing tumor genome architectures. *Bioinformatics* **19,** (2003).

38. Bafna, V. & Pevzner, P. A. Genome Rearrangements and Sorting by Reversals. *SIAM J. Comput.* **25,** 272–289 (1994).

39. Greenman, C. D. *et al.* Estimation of rearrangement phylogeny for cancer genomes. *Genome Res.* **22,** 346–361 (2012).

40. IBM. IBM ILOG CPLEX V12.1. (2009). at <ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf>

41. Almendro, V. *et al.* Inference of tumor evolution during chemotherapy by computational modeling and in situ analysis of genetic and phenotypic cellular diversity. *Cell Rep.* **6,** 514–27 (2014).

42. de Bruin, E. C., Taylor, T. B. & Swanton, C. Intra-tumor heterogeneity: lessons from microbial evolution and clinical implications. *Genome Med.* **5,** 101 (2013).

43. Klein, C. A. Selection and adaptation during metastatic cancer progression. *Nature* **501,** 365–72 (2013).

44. Bedard, P. L., Hansen, A. R., Ratain, M. J. & Siu, L. L. Tumour heterogeneity in the clinic. *Nature* **501,** 355–64 (2013).

45. Ding, L., Raphael, B. J., Chen, F. & Wendl, M. C. Advances for Studying Clonal

Evolution in Cancer. *Cancer Lett.* **340,** 212–219 (2013).

46. Malhotra, A. *et al.* Breakpoint profiling of 64 cancer genomes reveals numerous complex rearrangements spawned by homology-independent mechanisms. *Genome Res.* **23,** 762–776 (2013).

47. Emden, G., Eleftherios, K. & Stephen, N. Drawing graphs with dot. (2006). at <http://www.graphviz.org/Documentation/dotguide.pdf>

48. John, E., Emden, G., Eleftherios, K., Stephen, N. & Gordon, W. in *Graph Drawing Software* (eds. Jünger, M. & Mutzel, P.) 127–148 (Springer Berlin Heidelberg, 2004). doi:10.1007/978-3-642-18638-7_6

49. Stephens, P. J. *et al.* Massive genomic rearrangement acquired in a single catastrophic event during cancer development. *Cell* **144,** 27–40 (2011).