A global structural EM algorithm for a model of cancer progression

Ali Tofigh, Erik Sjölund, Mattias Höglund, and Jens Lagergren

Abstract

Cancer has complex patterns of progression that include converging as well as diverging progressional pathways. Vogelstein's path model of colon cancer was clearly a pioneering contribution to cancer research. Since then, several attempts have been made at obtaining mathematical models of cancer progression, devising training algorithms, and applying these to cross-sectional data.

Beerenwinkel *et al.* provided, what they coined, EM-like algorithms for Oncogenetic Trees (OTs) and mixtures of such. Given the small size of current and future data sets, it is important to minimize the number of parameters of a model. For this reason, also we focus on tree-based models and introduce Hidden-variable Oncogenetic Trees (HOTs). In contrast to OTs, HOTs allow for errors in the data and thereby provide more realistic modeling. We also design global structural EM algorithms for learning HOTs and mixtures of HOTs (HOT-mixtures). The algorithms are global in the sense that, during the M-step, they find a structure that yields a global maximum of the expected complete log-likelihood rather than merely one that improves it.

The algorithm for single HOTs performs very well on reasonable-sized data sets, while that for HOT-mixtures requires data sets of sizes obtainable only with tomorrows more cost efficient technologies. To facilitate analysis of complex cytogenetic data sets requiring more than one HOT, we devise a decomposition strategy based on Principal Component Analysis and train parameters on a colon cancer data set. The method so obtained is then successfully applied to kidney cancer.

1 Introduction

We view cells in cancer progression as progressing towards further malignancy by repeatedly being exposed to genetic or epigenetic aberrations that up-regulate, down-regulate, or dysregulate pathways. Thus, cancer progression is viewed as a walk through a set of states, each representing a set of affected pathways and the type of alterations they have been subjected to. This can be represented by a *a progression graph*, which is a directed graph where the vertices are states and the arcs represent possible transitions between them. Although a tumor is typically heterogeneous with respect to cell types, we make the common assumption that it is homogeneous; a proper discussion of this subject lies outside the scope of this

paper. Consider a situation where it is possible to repeatedly sample from the same tumor of a mouse and identify the malign cell types. Clearly, this would provide a path through our progression graph, and by concatenating paths obtained from different mice having the same cancer type the entire progression graph could conceivably be inferred. In this hypothetical situation, transition probabilities could also be estimated, which would provide a Markov chain. Unfortunately, the accessible biological samples typically do not comprise a time series for each tumor, but are cross-sectional, i.e., a data set is a collection of tumors that each have been removed from a different diseased individual after diagnosis.

In the near future, multiple types of high throughput (HTP) data will be available for large collections of tumors, providing great opportunities for state identification, and thereby, providing computational challenges for progression model inference. In this paper, we focus on cytogenetic data for colon and kidney cancer, mostly due to the availability of cytogenetic data for large numbers of tumors provided by the Mitelman database [17]. Rather than attempting to find a progression graph, we develop a tree-based model, since they have the significant advantage of having fewer parameters. It also turns out that these models allow for efficient algorithms. One of the main motivations for our models and inference methods is that they enable analysis of future HTP-data, which most likely will require the ability to handle large numbers of mutational events.

1.1 Mathematical models and algorithms

Vogelstein made a pioneering contribution to cancer research by proposing a path model for colon cancer. Since then, numerous examples of narrative models, often depicted with DAGs, e.g., [20], have been published. In an effort to provide mathematical models of cancer progression, Desper *et al.* [6] introduced the Oncogenetic Tree model where observable variables corresponding to aberrations are associated with vertices of a tree. They then proceeded to show that an algorithm based on Edmonds's optimum branching algorithm will, with high probability, correctly reconstruct an Oncogenetic Tree from sufficiently long series of data generated from it. In [7], another algorithm is described and shown to converge to an Oncogenetic Tree that generates a distribution close to the one generated by the true tree.

The Oncogenetic Tree model suffers from two problems: (1) monotonicity: an aberration associated with a child cannot occur unless the aberration associated with its parent has occurred, and (2) non-convergence: different progression paths cannot converge on the same aberration, as often is the case in tumor progression. In an attempt to remedy these problems, the Network Aberration Model was proposed [12, 18]. However, the computational problems associated with these network models are hard; for instance, no efficient EM algorithm for training is yet known. In another attempt, Beerenwinkel *et al.* used mixtures of Oncogenetic Trees to overcome the problem of non-convergence, but without removing the monotonicity and only obtaining an algorithm with an EM-like structure, which has not been proved to deliver a locally optimal maximum likelihood (ML) solution [1, 2, 19]. These mixture models were originally developed to model HIV evolution and were only later applied to model cancer progression.

It is customary to distinguish between EM algorithms and generalized EM algorithms, the difference being that in the M-step of the former, parameters are found that maximize the expected complete log-likelihood, whereas in the latter, parameters are found that merely improve it. As Friedman notes in his article on the Bayesian Structural EM algorithm [10], the same distinction can be made regarding the maximization over structures. Clearly, it would be convenient to use the same terminology for structural EM algorithms as for ordinary EM algorithms. However, for structural EM algorithms, the distinction is often not made, and even researchers that consider themselves experts in the field seem to be unaware of it. For this reason, we define *global* structural EM algorithms to be EM algorithms that in the M-step find a structure yielding a global maximum of the expected complete log-likelihood (rather than merely improving the expected complete log-likelihood).

In the learning literature, there are several previous results on learning trees and global structural EM algorithms. Chow and Lieu considered trees where the vertices were associated with observable variables and gave an efficient algorithm for finding a globally optimal ML solution [4]. Subsequently, Meila *et al.* presented a global structural EM algorithm for finding the ML mixture of trees [16], as well as MAP solutions with respect to various priors. Friedman *et al.* [11] described a global structural EM algorithm for phylogenetic trees. It is interesting, in relation to the present result, to note that Friedman *et al.* consider phylogenetic trees, i.e., trees with observable variables associated to leaves and hidden variables associated to the internal vertices, and where, moreover, a reversible probabilistic model relates any pair of variables associated with neighboring vertices. Solving the maximum spanning tree problem for a weighted graph is a main component of all these algorithms.

We present the Hidden-variable Oncogenetic Tree (HOT) model where a hidden and an observable variable are associated with each vertex of a rooted directed tree. The value of the hidden variable indicates whether the tumor progression has reached the vertex (a value of one means that cancer progression has reached the vertex and zero that it has not), while the value of the observable variable indicates whether a specific aberration has been detected in HTP-data (a value of one represents detection and zero the opposite). This interpretation provides several relations between the variables in a HOT that are specified in the formal definition of our model. An asymmetric relation is required between the hidden variables associated with the two endpoints of an arc of the directed tree. Because of the asymmetry, the global structural EM algorithm that we derive for the HOT ML problem can, in contrast to the above mentioned algorithms, not be based on a maximum spanning tree algorithm, and is instead based on the optimal branching algorithm [3, 15, 21]. Having so rectified the monotonicity problem, we proceed to obtain a model allowing for a higher degree of convergence by introducing mixtures of HOTs (HOT-mixtures) and, in contrast to Beerenwinkel *et al.*, we derive a proper structural EM algorithm for training these.

We focus on tree models for two reasons: (1) there is a global structural EM algorithm for inference from cross-sectional data and (2) tree models have few parameters. The latter is very important due to the relatively small number of data points available, both in data sets today and in those of the future, compared to the number of mutational events under consideration. It has been observed that cancer progression paths can diverge as well as

converge. In one form of cancer two tumors having different possibly disjoint sets of aberrations such as $\{1,2\}$ and $\{3,4\}$ may both obtain the aberration 5, i.e., they converge in the sense that they both obtain the same aberration. It is also possible to have convergence when two tumors with different sets of aberrations both make transitions to the same set of aberrations. Divergence is possible when two different tumors having progressed along the same path of states up to some point in the next step acquire different aberrations. The underlying tree structure of a HOT allows for divergence and convergence, and HOT-mixtures allow for convergence to an even greater extent. Again, in our HOT model, hidden variables model the cancer progression and observable variables correspond to detection of progression in data. So, in contrast to Oncogenetic trees and mixtures of such, HOTs and HOT-mixtures can handle cases where in some tumors a subset of aberrations are undetected in HTP-data.

In Section 2, we show how to model cancer progression by using HOTs and HOT-mixtures. In section 3, this modeling methodology is applied to cytogenetic copy number aberration (CNA) data for colon and kidney cancer.

2 HOTs and the novel global structural EM algorithm

This section contains four subsections. In the first, we introduce the HOT model and compare it to the OT model. Subsection 2.2 contains a description of our EM algorithm for training HOTs. In subsection 2.3, we show how to compute certain probabilities that are required during training. Finally, an EM algorithm for training HOT-mixtures is described in subsection 2.4.

2.1 Hidden-variable Oncogenetic Trees

We will denote the set of observed data points D and an individual data point X. In Section 3, we will apply our methods to CNA, i.e., a data point will be a set of observed CNA, but in general, more complex events can be used.

A *rooted directed tree* T consists of a set of vertices, denoted $V(T)$ and a set of arcs denoted $A(T)$. An arc $\langle u, v\rangle$ is directed from the vertex u called its *tail* towards the vertex v called its *head*. If there is an arc with tail p and head u in a directed tree T, then p is called the parent of u in T and denoted $p(u)$ (the tree T will be clear from context).

An OT is a rooted directed tree where there is an aberration associated with each vertex and a probability associated with each arc. One can view an OT as generating a set of aberrations by first visiting the root and then continuing towards the leaves (preorder) visiting each vertex with the probability of its incoming arc if the parent has been visited, and with probability zero if the parent has not been visited. Finally, the result of the progression is the set of aberrations associated with the visited vertices.

In Figure 1(b), an OT for CNA is depicted. It can generate the following sets of CNAs: $\emptyset$, $\{-3p\}$, $\{-3p,-4p\}$, $\{-3p,+Xp\}$, $\{-3p,-4p,+Xp\}$, $\{+17q\}$, $\{-3p,+17q\}$, $\{-3p,-4p,+17q\}$, $\{-3p,+Xp,+17q\}$, and $\{-3p,-4p,+Xp,+17q\}$ (all these aberrations are written in the standard notation for CNAs in cytogenetic data, i.e., each represents a duplication (+) or

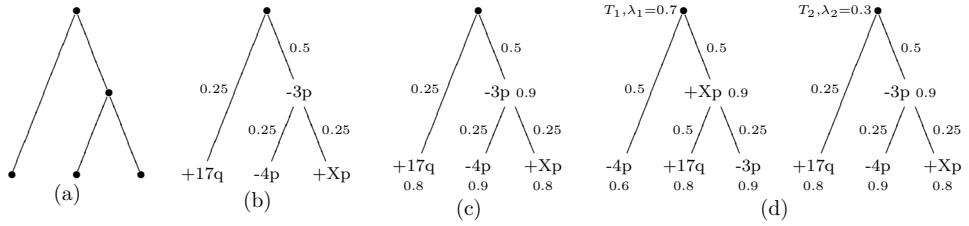

Figure 1: **(a)** *A rooted directed tree with the root at the top. All arcs are directed downwards, i.e., away from the root.* **(b)** *An OT with probabilities associated with arcs and CNAs associated with vertices.* **(c)** *A HOT with probabilities associated with arcs (indicating the probability that the hidden variable associated with the head of the arc receives the value* 1 *conditioned that the hidden variable associated with the tail has this value), and CNAs as well as probabilities associated with vertices (indicating the probability that the observable variable associated with the vertex receives the value* 1 *conditioned that the hidden variable associated with the vertex has received this value).* **(d)** *A HOT-mixture consisting of two HOTs. The mixing probability for* T_1 *is* 0.7 *and that for* T_2 *is* 0.3. *So with probability* 0.7 *a synthetic tumor is generated from* T_1 *and otherwise one is generated from* T_2.

deletion (-) of a specific chromosomal region). Notice that an aberration associated with a vertex cannot occur unless the aberration associated with its parent has occurred. For instance, the set $\{+Xp, +17q\}$ cannot be generated by the OT in Figure 1(b). In a data-modeling context, this is highly undesirable, since data is typically noisy and whatever mutational events we are modeling some of those that have occurred are likely to have gone undetected. Our HOT model does not suffer from this problem.

A Hidden-variable Oncogenetic Tree (HOT) is a directed tree where there is an aberration associated with each vertex and a probability associated with each arc, exactly as in a OT. Moreover, in contrast to the OT, there is also a probability associated with each vertex. One can view the HOT as generating data by first allowing cancer progression to reach a subset of the vertices of the tree, exactly as in an OT, i.e., based on the probabilities associated with the arcs. In the HOT, however, an aberration associated with a vertex reached by the progression process is not automatically generated, instead it is generated with the probability associated with that vertex. As an example consider the HOT illustrated in Figure 1(c). The probability that it generates the set $\{+Xp, +17q\}$ is $0.25{\cdot}0.8{\cdot}0.5{\cdot}0.1{\cdot}0.25{\cdot}0.8{\cdot}(0.75{+}0.25{\cdot}0.1) = 0.00155$.

We will now give a formal definition of HOTs. Notice that the probabilities associated with edges in the description above are the conditional probabilities in (4) and those associated with vertices are the conditional probabilities in (5). A Hidden-variable Oncogenetic Tree (HOT) is a pair $\mathcal{T} = (T, \Theta)$ where:

1. T is a rooted directed tree and Θ consists of two conditional probability distributions (CPDs), $\theta_X(u)$ and $\theta_Z(u)$, for each vertex u;

2. two random variables are associated with each vertex $u \in V(T)$: an observable variable $X(u)$ and a hidden variable $Z(u)$, each assuming the values 0 or 1,

3. the hidden variable associated with the root, $Z(r)$, always assumes the value 1,

4. for each non-root vertex u of $V(T)$, $\theta_Z(u)$ is a conditional probability distribution on $Z(u)$ conditioned by $Z(p(u))$ satisfying $\Pr[Z(u) = 1|Z(p(u)) = 0] = 0$, and

5. for each non-root vertex u of $V(T)$, $\theta_X(u)$ is a conditional probability distribution on $X(u)$ conditioned by $Z(u)$ satisfying $\Pr[X(u) = 1|Z(u) = 0] = 0$.

For practical reasons, in the implementation of the algorithm, we use the condition $\Pr[Z(u) = 1|Z(p(u)) = 0] = \epsilon_Z$, where ϵ_z is a small value, rather than the strict condition in (4). The motivation is basically the same as for using so called pseudo-counts [8]. Namely, once a parameter receives the value 0 in an EM algorithm for training, it will subsequently not be changed. For modeling reasons, we use the condition $\Pr[X(u) = 1|Z(u) = 0] = \epsilon_X$ for some small ϵ_X rather than the condition stated in (5).

In Subsection 3, when modeling a collection of tumors represented by CNAs, we will number the CNAs $1, \ldots, n$ and also use these numbers to represent the non-root vertices, and we will consider a CNA i to have happened if and only if $X(i) = 1$, i.e., the final set of aberrations generated is $\{i : X(i) = 1\}$.

It is also possible to have CPDs where $X(u)$ and $Z(u)$ depend on both $X(p(u))$ and $Z(p(u))$ and even to let $X(u)$ depend on all three of $Z(u)$, $X(p(u))$, and $Z(p(u))$. We do not cover these cases in the following text, but our arguments can easily be extended to also cover these.

2.2 The novel global structural EM algorithm for HOTs

When viewing probabilistic models as generating data, the model-training problem can be cast as an optimization problem where the goal is to find the maximum likelihood solution, i.e., the model that with the highest probability generates the observed data. This optimization problem is often solved using an Expectation Maximization (EM) algorithm [5], which is not guaranteed to deliver a globally optimal solution but used to obtain locally optimal ones.

The EM theory shows that given a current solution, another solution with higher likelihood can be found by maximizing the so-called expected complete log-likelihood or the Q-term. Friedman *et al.* [11] extended the use of EM algorithms from the standard parameter estimation to also finding an optimal structure. In their case, the probabilistic model was reversible which makes it possible to maximize the expected complete likelihood over all trees, using a maximum spanning tree algorithm. In our case, the pair-wise relations between hidden variables are asymmetric. However, as shown below, the maximization of the expected complete log-likelihood can in our case be solved using Edmonds's optimal branching algorithm. Tarjan's variation of Edmonds's algorithm runs in quadratic time [3, 15, 21].

The *weighted* expected complete log-likelihood function will be useful when treating HOT-mixtures. We introduce it already here and also show how to maximize it. The expected complete log-likelihood of a HOT $\mathcal{T}'$ with respect to another HOT $\mathcal{T}$, weighted by a function f, and with our observed variables as parameters, is defined as

$$Q_f(\mathcal{T}';\mathcal{T}) = \sum_{X\in D}\sum_{Z} f(X)\Pr[Z|X,\mathcal{T}]\log\Pr[Z,X|\mathcal{T}']. \tag{1}$$

We now show that if f can be evaluated in constant time, then the HOT $\mathcal{T}'$ that maximizes (1) can be found in time $O(n^2)$, where n is the number of aberrations or vertices.

Following the standard derivation of an EM algorithm, it can be shown that $Q_f(\mathcal{T}';\mathcal{T})$ equals

$$\sum_{\langle u,v\rangle\in A(T')}\sum_{a,b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(v)=a,Z(u)=b|X,\mathcal{T}]\log\Pr[Z(v)=a|Z(u)=b,\theta'_Z(u)]$$
$$+\sum_{\langle u,v\rangle\in A(T)}\sum_{\sigma,a\in\{0,1\}}\sum_{X\in D:X(u)=\sigma} f(X)\Pr[Z(v)=a|X,\mathcal{T}]\log\Pr[X(v)=\sigma|Z(v)=a,\theta'_X(u)].$$

As long as the directed tree T' is fixed, the standard EM methodology (see for instance [8]) can be used to find the Θ' that maximizes $Q_f(T',\Theta';\mathcal{T})$, as follows. First, let

$$A_u(a,b) = \sum_{X\in D} f(X)\Pr[Z(u)=a,Z(p'(u))=b|X,\mathcal{T}] \tag{2}$$

and

$$B_u(\sigma,a) = \sum_{X\in D:X(u)=\sigma} f(X)\Pr[Z(u)=a|X,\mathcal{T}]. \tag{3}$$

Then the Θ' that for a fixed T' maximizes $Q_f(\mathcal{T}';\mathcal{T})$ (i.e. $Q_f(T',\Theta';\mathcal{T})$) is given by letting

$$\Pr[Z(u)=a|Z(p'(u))=b,\theta'_Z(u)] = A_u(a,b)/(\sum_{a\in\{0,1\}} A_u(a,b))$$

and

$$\Pr[X(u)=\sigma|Z(u)=a,\theta'_Z(u)] = B_u(\sigma,a)/(\sum_{\sigma\in\{0,1\}} B_u(\sigma,a)).$$

In the next Section 2.3, we will describe how the probabilities on the right hand sides of (2) and (3) can be computed. The time required for computing all such probabilities will turn out to be no more than $O(|D|\cdot n^2)$, where n is the number of aberrations.

For each arc $\langle p,u\rangle$ of T', using the CPDs defined above, we define the weight of the arc, specific to this tree to be

$$\sum_{a,b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(u)=a,Z(p'(u))=b|X,\mathcal{T}]\log\Pr[Z(u)=a|Z(p'(u))=b,\theta'_Z(u)]$$
$$+\sum_{b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(u)=a|X,\mathcal{T}]\log\Pr[X(u)|Z(u)=a,\theta'_X(u)].$$

We now make two important observations from which it follows how to maximize the weighted expected complete log-likelihood over all directed trees. First, notice that if two directed trees T' and T'' have a common arc $\langle p, u\rangle$, then this arc has the same weight in these two trees. This allows us to define a complete directed and arc-weighted graph D (i.e., a combinatorial structure with a set of vertices and an arc in each direction between any two vertices) with the same vertex set as the tree T, and define the weight of the arc $\langle u, v\rangle$ in this directed graph to be the same as in any directed tree containing the arc.

An optimal arborescence of a directed graph is a rooted directed tree on the same set of vertices as the directed graph that has exactly one directed path from one specific vertex called the root to any other vertex and, moreover, has maximum arc weight sum among all such rooted directed trees. Now we are in position to conclude that Edmonds's optimal branching algorithm (of which a variation can produce an optimal arborescence) can be used to maximize the weighted expected complete log-likelihood. For any branching T' of D, the sum of the weights of its arcs equals by construction the maximum value of $Q_f(T', \Theta'; \mathcal{T})$ for any Θ'. From this follows that, a (spanning) directed tree T' is an optimal branching of D if and only if T' maximize the Q_f term. So applying Tarjan's variation of Edmonds's algorithm [3, 15, 21] to D gives the desired directed tree. In the next subsection, we show how to compute the probabilities required in (2) and (3), thereby, complete the description of our model-training algorithm for HOTs.

2.3 Computing the required probabilities

The most basic computation for a HOT $\mathcal{T} = (T, \Theta)$ is computing the probability that an observation X is generated from $\mathcal{T}$, i.e., $\Pr[X|\mathcal{T}]$. This probability as well as the probabilities $\Pr[Z(u) = a, X|\mathcal{T}]$ and $\Pr[Z(u) = a, Z(p(u)) = b, X|\mathcal{T}]$ can be computed in linear time using dynamic programming, i.e., a procedure very similar to the pruning algorithm used to compute likelihoods of phylogenetic trees [9]. Doing so for all vertices u can, hence, be done in time $O(n^2)$. Using the above probabilities, we can in linear time compute $\Pr[Z(u) = a|X, \mathcal{T}]$ and $\Pr[Z(u) = a, Z(p(u)) = b|X, \mathcal{T}]$ for all vertices u. Finally, using the so computed probabilities, the probability $\Pr[Z(u) = a, Z(v) = b|X, \mathcal{T}]$ can then be computed using techniques analogous to those appearing in [11].

2.4 HOT-mixtures

In the previous section, we considered a HOT $\mathcal{T} = (T, \Theta)$. We will now extend the model to HOT-mixtures by including an initial random choice of one out of several HOTs and letting the final outcome be generated by the chosen HOT. We will also obtain an EM based model-training algorithm for HOT-mixtures by showing how to optimize expected complete log-likelihoods for HOT-mixtures. Formally, we will use k HOTs $\mathcal{T}_1, \ldots, \mathcal{T}_k$ and a random mixing variable I taking on values in $1, \ldots, k$. The probability that I gets the value i is denoted λ_i and $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a vector of parameters of the model, in addition to those

of the HOTs ($\lambda_1, \ldots, \lambda_k$ are constrained to sum to 1). The following notation is convenient

$$\gamma_i(X) = \Pr[I = i|X, M] = \frac{\lambda_i \Pr[X|\mathcal{T}_i]}{\sum_{j\in[k]} \lambda_j \Pr[X|\mathcal{T}_j]}.$$

For a HOT-mixture, the expected complete log-likelihood can be expressed as follows

$$\sum_{X\in D} \sum_{Z,I} \Pr[Z, I|X, M] \log \Pr[Z, I, X|M']. \tag{4}$$

Using standard EM methodology, it is possible to show that (4) can be maximized by independently maximizing

$$\sum_{i\in[k]} \sum_{X\in D} \gamma_i(X) \log(\lambda'_i) \tag{5}$$

and, for each $i = 1, \ldots, k$, maximizing

$$\sum_{X\in D} \sum_{Z} \Pr[Z|X, \mathcal{T}_i]\gamma_i(X) \log(\Pr[Z, X|\mathcal{T}'_i]) \tag{6}$$

Finding a $\lambda' = \lambda'_1, \ldots, \lambda'_k$ maximizing (5) is straightforward (see for instance [8]) and, for each $i = 1, \ldots, k$, finding a $\mathcal{T}'_i$ maximizing (6) can be done as described in the previous subsections.

3 Results

In this section, we report results obtained by applying our algorithms to synthetic data as well as cytogenetic cancer data. For ease of notation, we will denote the parameters of the model as follows:

$$\begin{aligned}
p_z(u) &= \Pr[Z(u) = 1|Z(p(u)) = 1] \\
p_x(u) &= \Pr[X(u) = 1|Z(u) = 1] \\
e_z(u) &= \Pr[Z(u) = 1|Z(p(u)) = 0] \\
e_x(u) &= \Pr[X(u) = 1|Z(u) = 0].
\end{aligned}$$

We will collectively call the three parameters, $(1 - p_x)$, e_z, and e_x, the *error parameters*.

In the standard version of the EM algorithm, four parameters are associated with each edge of a HOT. In order to reduce the total number of parameters, it is possible to let some of the four parameters be global instead. For example, in the case of e_x, this means that we would require that $e_x(u) = e_x(u')$ for all pairs of vertices u and u'. Ideally, we would like to let all the error parameters be global. However, for technical reasons, requiring that e_z be global makes it impossible to derive an EM algorithm. Therefore, we will distinguish between two different versions of the algorithm: one with *free parameters* and one with *global*

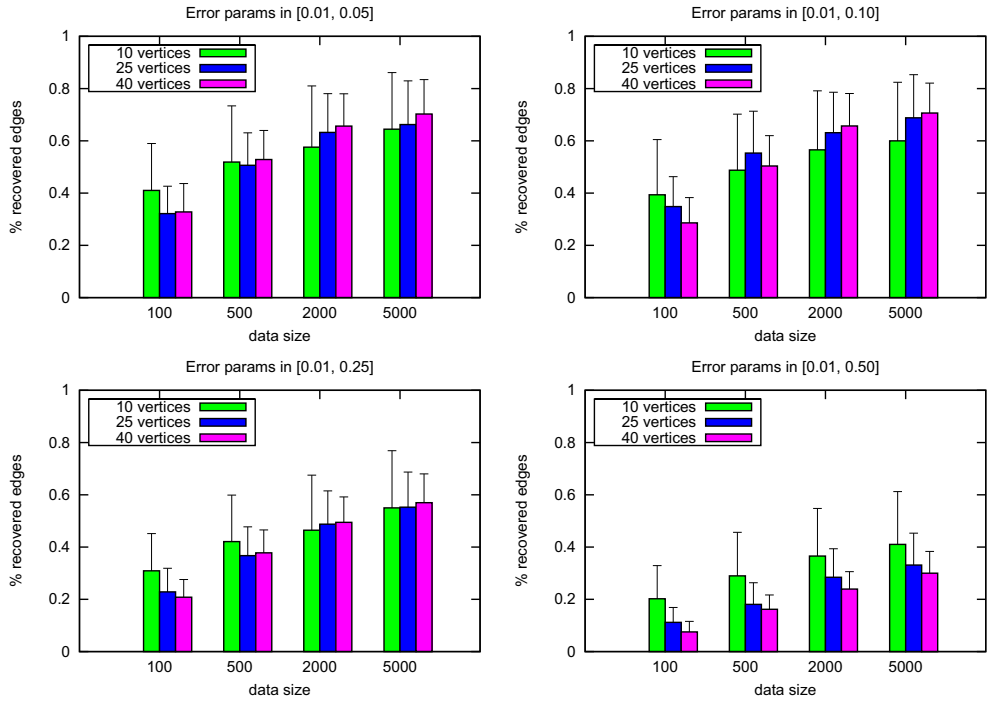

Figure 2: Histograms showing the mean percentage of edges that were correctly recovered by the algorithm for the free parameter case together with errorbars showing one standard deviation.

parameters. The free parameter version then corresponds to the standard EM algorithm, while the global parameter version corresponds to letting $(1 - p_x)$ and e_x be global. When evaluating the global parameter version of the algorithm using synthetic data, we will follow the convention of letting all three error parameters be global when generating data.

Other conventions used for all the tests described here include the following. Unless stated otherwise, we enforce an upper limit of 0.5 on e_z and e_x. Also, when running the algorithm on a dataset, we first run the algorithm on a set of randomly generated start HOTs or start HOT-mixtures for 10 iterations. The HOT or HOT-mixture that resulted in the best likelihood is then run until convergence. Unless stated otherwise, the number of start trees or mixtures is 100.

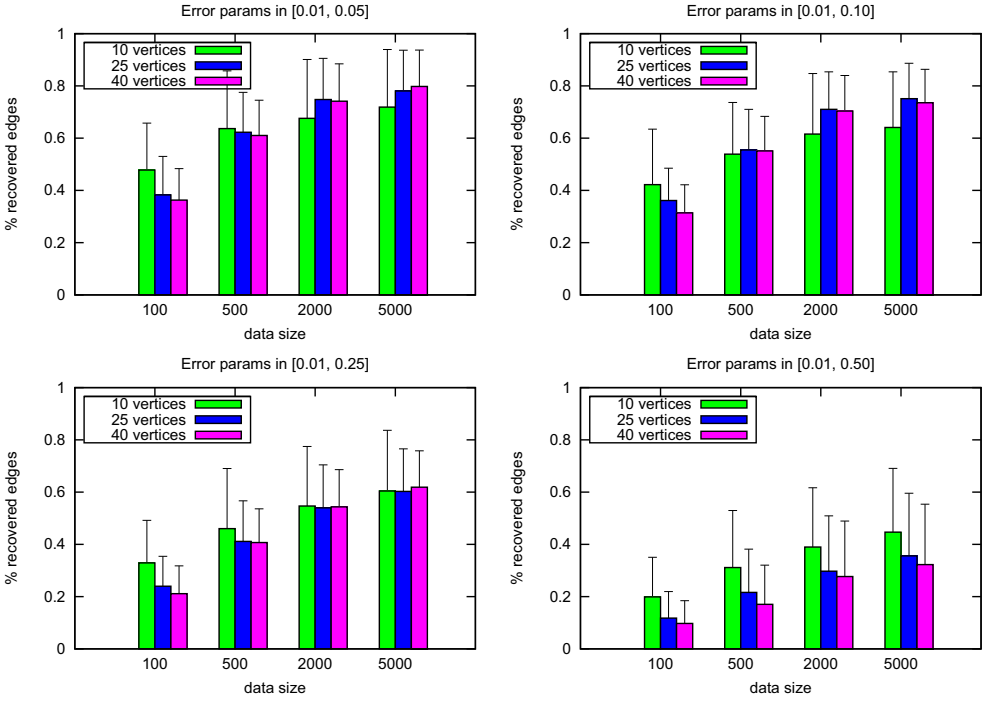

Figure 3: Histograms showing the mean percentage of edges that were correctly recovered by the algorithm for the global parameter case together with errorbars showing one standard deviation.

3.1 Tests on Synthetic Datasets

3.1.1 Single HOTs

In order to test the algorithm's ability to recover a HOT from data, we generated random HOTs with different sizes and parameters. We then generated data from these HOTs and attempted to recover the HOTs using the hotmix algorithm. The sizes of the HOTs were fixed at 10, 25, and 40 vertices. The parameters on the edges, i.e., the probabilities p_z, p_x, e_z, and e_x, were chosen uniformly in the intervals

$$p_z \in [0.1, 1.0], \tag{7}$$

$$(1 - p_x), e_x, e_z \in [0.01, q], \tag{8}$$

where $q \in \{0.05, 0.10, 0.25, 0.50\}$. For each combination of possible sizes and values for q, 100 HOTs were generated for a total of $3 \times 4 \times 100 = 1200$ HOTs. Data was generated from each HOT with 100, 500, 2000, and 5000 datapoints. Each dataset was then passed to the algorithm and the resulting HOT was compared to the original HOT. The result of this comparison can be seen in Figure 2. An edge of the original HOT connecting one specific aberration to another is considered to have been correctly recovered if the HOT obtained from the algorithm connects the same two aberrations in the same direction.

We also tested how the reduction of parameters affected the results by generating HOTs with global error parameters. This is the so-called "global parameters" case as described in the introduction. We then applied the relevant version of the algorithm, and the results can be seen in Figure 3. As shown by the figures, there is a slight improvement when the number of parameters is reduced.

We also compared the performance of our algorithms with that of *Mtreemix* by Beerenwinkel *et al* [2]. The generated data from our single HOTs were passed to Mtreemix and the same criteria as above were used to detect correctly recovered edges (no special options were set when running Mtreemix on data generated with global parameters since no distinction between global and free parameters can be made on oncogenetic trees) Figure 4 and 5 show the results. As can be seen, Mtreemix outperforms our methods when the HOTs and the error parameters are small, and our algorithms outperform Mtreemix significantly as the HOTs or error parameters become larger.

3.1.2 HOT Mixtures

We also tested the ability of the algorithm to recover a mixture of two HOTs. The sizes of the HOTs were set at either 10 or 25 vertices (i.e. a total of 18 or 48 edges). The error parameters, which were global, were chosen randomly from a uniform distribution on the interval $[0.01, q]$ where $q \in \{0.05, 0.10, 0.25\}$. Three different mixture distributions on the HOTs were also tested.

When measuring the number of correctly recovered edges, the following procedure was used. Each HOT produced from the algorithm was compared to each HOT from which the data was generated, and the number of correctly recovered edges was noted. The best way

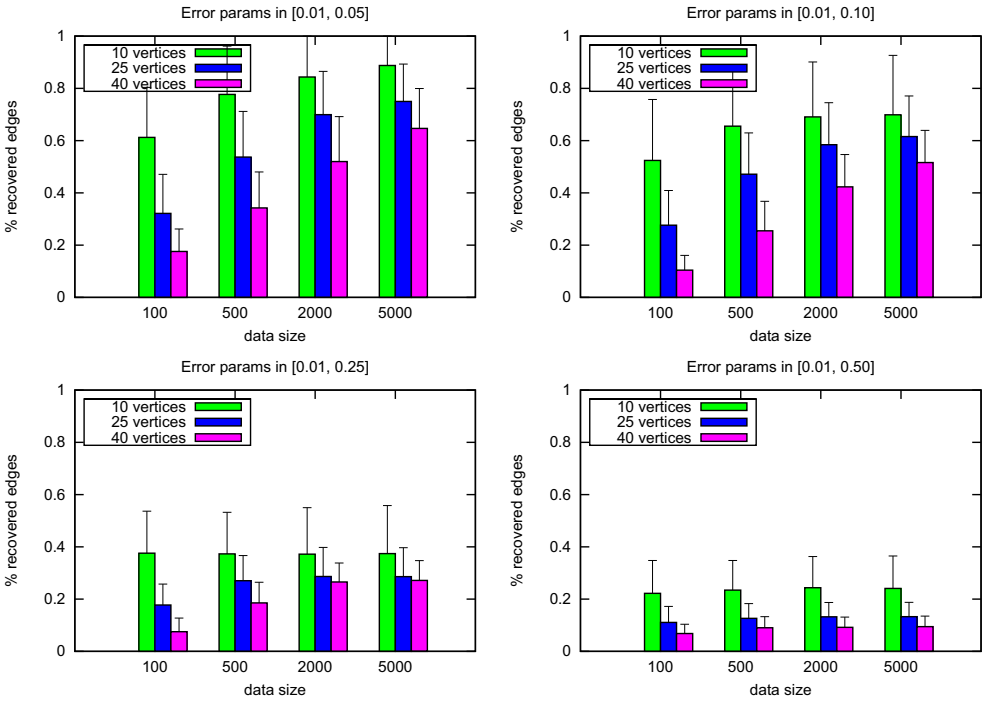

Figure 4: Histograms showing the mean percentage of edges that were correctly recovered by Mtreemix together with errorbars showing one standard deviation. The data was the same as those used in Figure 2.

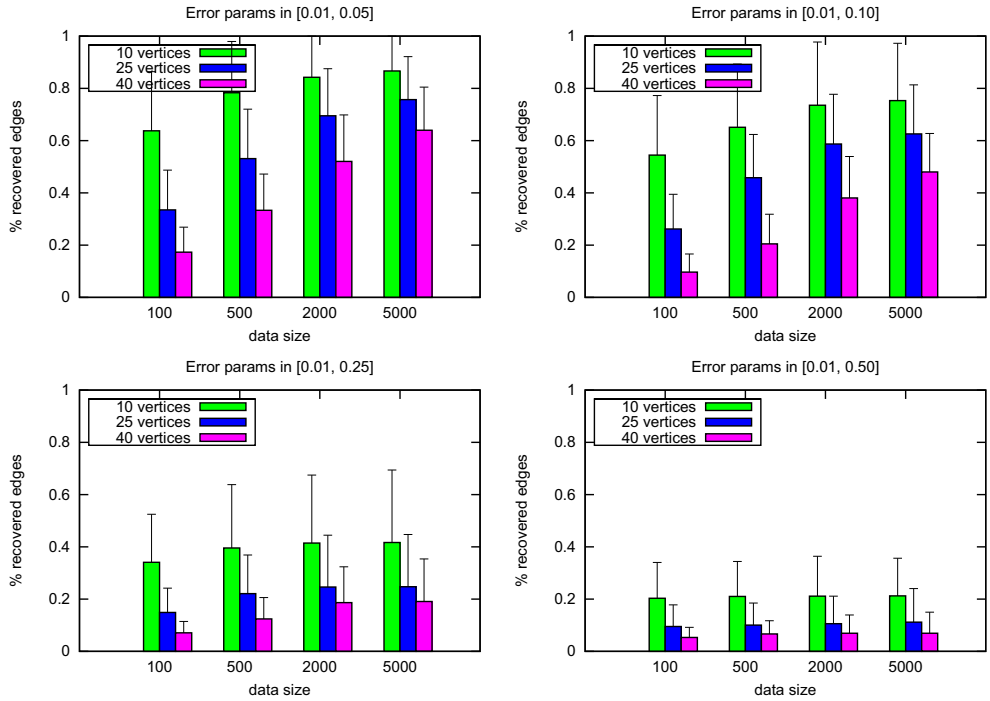

Figure 5: Histograms showing the mean percentage of edges that were correctly recovered by Mtreemix together with errorbars showing one standard deviation. The data was the same as those used in Figure 3.

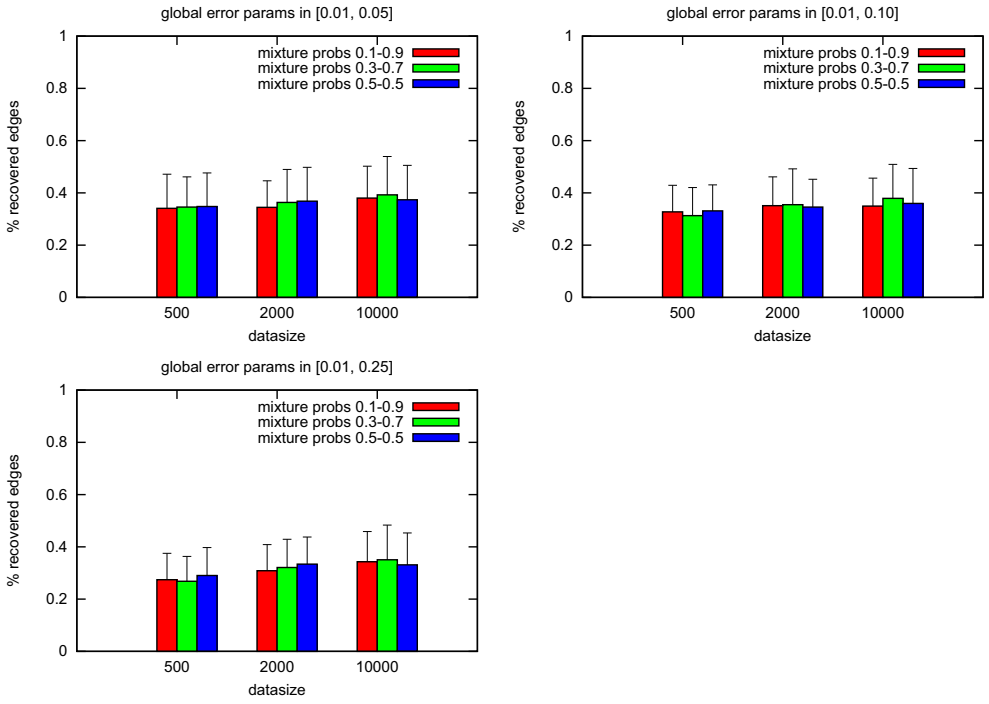

Figure 6: Histograms showing the mean percentage of edges that were correctly recovered for mixtures of two HOTs with 10 vertices each. The errorbars indicate one standard deviation. Each bar represents 100 mixtures.

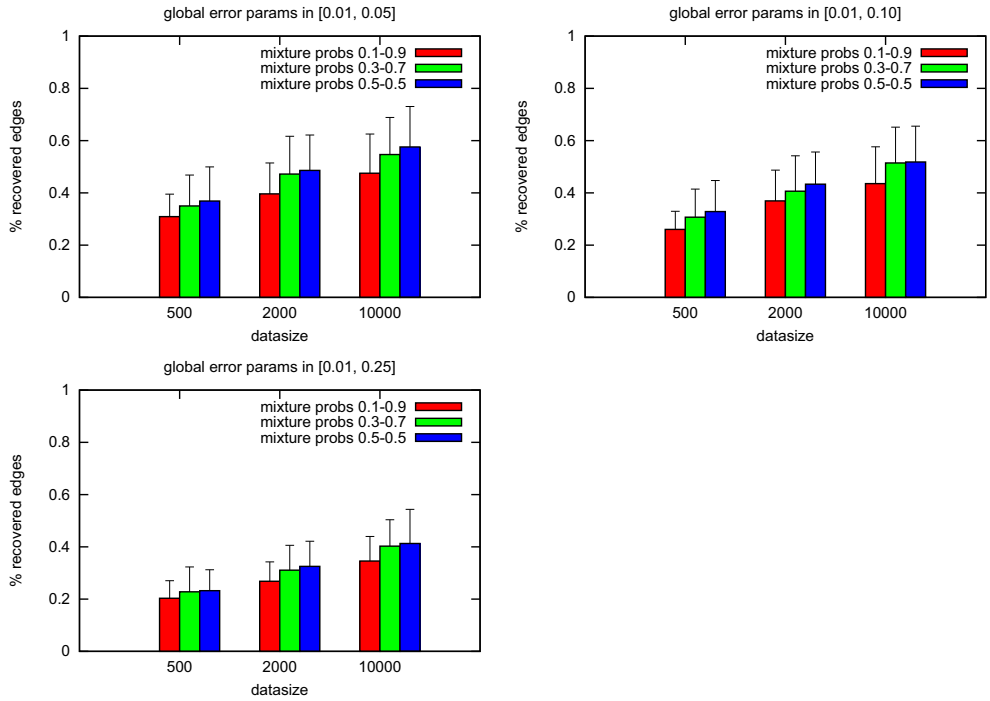

Figure 7: Histograms showing the mean percentage of edges that were correctly recovered for mixtures of two HOTs with 25 vertices each. The errorbars indicate one standard deviation. Each bar represents 100 mixtures.

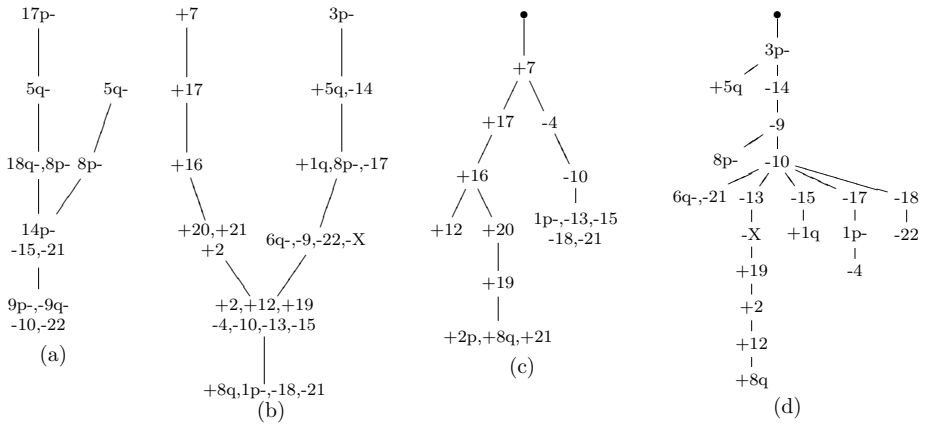

Figure 8: **HOTs obtained from RCC data.** *(a) shows an adapted version of the pathways for CC data published in [13]. (b) is a figure adapted from [14] showing the pathways obtained from statistical analysis of RCC data. (c) and (d) are the HOTs we obtained from the RCC data using only aberrations on the left and right pathways in (b), respectively.*

of matching the two HOTs produced from the algorithm with the two original HOTs was then determined. The result can be seen in Figure 6 and 7.

For the case with 25 vertices, two features can clearly be distinguished: the results improve as the size of the data increases, and the algorithm performs better when the HOTs have equal probability in the mixture.

3.2 Tests on Cancer Data

Our cytogenetic data for colon (CC) and kidney (RCC) cancer consist of 512 and 998 tumors, respectively. The data consist of measurements on 41 common aberrations (18 gains, 23 losses) for CC and 28 (13 gains, 15 losses) for RCC. The data have previously been analyzed in [13] and [14] resulting in suggested pathways of progression. These analyses were based on Principal Component Analysis (PCA) performed on correlations between aberrations and a statistical measure called *time of occurrence* (TO) that is a measure on how early or late an aberration occurs during progression. The aberrations were then clustered based on the PCA and each cluster was manually formed into a pathway (based on PCA and TO). One advantage of our approach is that we are able to replace the manual curation by automated computational steps. Another advantage is that our models assign probabilities to data and the different models can therefore be compared objectively.

We expect the parameters $e_z(u) = \Pr[Z(u) = 1|Z(p(u)) = 0]$ and $e_x(u) = \Pr[X(u) = 1|Z(u) = 0]$ to be small in real data. We obtained the n most correlated aberrations in our CC data, for $n \in \{4, \ldots, 11\}$, and tested different upper limits on e_z and e_x. The best

correspondence to previously published analyses of the data was found when $e_z(u) \leq 0.25$ and $e_x(u) \leq 0.01$ with the number of *bad edges* given in the table below. A bad edge is one that contradicts the partial ordering given by the pathways described in [13], of which the relevant part is shown in the Figure 8(a).

size	4	5	6	7	8	9	10	11
bad edges	0	0	0	0	2	2	3	2

Having found upper limits that work well on the CC data, we applied the algorithm with these upper bounds to the RCC data. The earlier analyses in [14] strongly suggests that two HOTs are required to model the RCC data. Given that our mixture model, from synthetic data tests, appears to require substantially more data points to recover the underlying HOTs in a satisfactory manner, we used the results of the analysis in [14] to divide the aberrations into two (overlapping) clusters for which we created HOTs separately. These HOTs can be seen in Figure 8(c) and 8(d) and they show very good agreement to the pathways from [14] shown in Figure 8(b). For instance, each root-to-leaf path in the HOT of Figure 8(c) agrees perfectly with the pathway shown in Figure 8(b).

References

[1] N. Beerenwinkel, J. Rahnenfuhrer, M. Daumer, D. Hoffmann, R. Kaiser, J. Selbig, and T. Lengauer. Learning multiple evolutionary pathways from cross-sectional data. *J Comput Biol*, 12(6):584–598, Jul 2005.

[2] N. Beerenwinkel, J. Rahnenfuhrer, R. Kaiser, D. Hoffmann, J. Selbig, and T. Lengauer. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*, 21(9):2106–2107, 2005.

[3] P. Camerini, L. Fratta, and F. Maffioli. The k best spanning arborescences of a network. *Networks*, 10(2):91–110, 1980.

[4] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[6] R. Desper, F. Jiang, O.P. Kallioniemi, H. Moch, C.H. Papadimitriou, and A.A. Schaffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *J Comput Biol*, 6(1):37–51, 1999.

[7] R. Desper, F. Jiang, O.P. Kallioniemi, H. Moch, C.H. Papadimitriou, and A.A. Schaffer. Distance-based reconstruction of tree models for oncogenesis. *J Comput Biol*, 7(6):789–803, 2000.

[8] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, Cambridge, 1998.

[9] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA, 2004.

[10] N. Friedman. The bayesian structural em algorithm. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 129–138. Morgan Kaufmann, 1998.

[11] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *J Comput Biol*, 9(2):331–353, 2002.

[12] M. Hjelm, M. Höglund, and J. Lagergren. New probabilistic network models and algorithms for oncogenesis. *J Comput Biol*, 13(4):853–865, May 2006.

[13] M. Höglund, D. Gisselsson, G.B. Hansen, T. Säll, F. Mitelman, and M. Nilbert. Dissecting karyotypic patterns in colorectal tumors: Two distinct but overlapping pathways in the adenoma-carcinoma transition. *Cancer Res*, 62:5939–5946, 2002.

[14] M. Höglund, D. Gisselsson, M. Soller, G.B. Hansen, P. Elfving, and F. Mitelman. Dissecting karyotypic patterns in renal cell carcinoma: an analysis of the accumulated cytogenetic data. *Cancer Genetics and Cytogenetics*, 153(1):1–9, 2004.

[15] R.M. Karp. A simple derivation of edmond's algorithm for optimum branching. *Networks*, 1(265-272):5, 1971.

[16] M. Meila and M.I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(1):1–48, 2000.

[17] F. Mitelman, B. Johansson, and F. Mertens. Mitelman database of chromosome aberrations in cancer, 2004. http://cgap.nci.nih.gov/Chromosomes/Mitelman.

[18] M.D. Radmacher, R. Simon, R. Desper, R. Taetle, A.A. Schaffer, and M.A. Nelson. Graph models of oncogenesis with an application to melanoma. *J Theor Biol*, 212(4):535–48, Oct 2001.

[19] J. Rahnenfuhrer, N. Beerenwinkel, W.A. Schulz, C. Hartmann, A. von Deimling, B. Wullich, and T. Lengauer. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. *Bioinformatics*, 21(10):2438–2446, May 2005.

[20] P.T. Simpson, J.S. Reis-Filho, T. Gale, and S.R. Lakhani. Molecular evolution of breast cancer. *J Pathol*, 205(2):248–254, Jan 2005.

[21] R.E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–36, 1977.