DISCOVERY OF MUTATED COLLECTIONS OF GENES ASSOCIATED WITH SURVIVAL IN CANCER USING LOCAL SEARCH

Patrick Clay
Undergraduate Thesis
Math and Computer Science, Sc. B.
Brown University 2013

Advisor: Eli Upfal
Reader: Fabio Vandin

Abstract

In studying the correlations between genetic mutations and cancer development, the relationship between mutations and patient survival is one of the most significant from both academic and clinical perspectives. Survival analysis being an established field, assessing the correlation between any given mutation or collection of mutations and survival is a straightforward process. This paper seeks to develop techniques to use survival analysis to identify *de novo* collections of mutations correlated with low survival. This requires exploring a very large search space of collections of genes, which cannot be effectively enumerated with computational resources. Instead local search in the form of steepest ascent hill climbing and a more generalized branching algorithm will be used to iteratively construct strongly correlated sets. Theoretical analysis of the runtimes demonstrate the feasibility of these algorithms under some limited assumptions. Empirical results demonstrate the ability of these algorithms to recover sets strongly correlated with survival. Some evidence is presented indicating that these techniques, while somewhat effective at present, will be significantly more so when cancer genomic datasets reach two to three times there current size.

CONTENTS

1 INTRODUCTION

Somatic mutations, occurring throughout one's lifetime, can mutate healthy cells to cancerous ones. These mutations vary greatly between individuals, even those with the same variety of cancer. This is because most mutations are *passenger* mutations and do not strongly increase risk of cancer and are considered, adding noise to genomic data. More importantly the *driver* mutations do not all need to affect the same gene, they simply need to affect any member of a collection of genes that in some way inhibits the production or survival of cancerous cells. Finding *de novo* collections of such genes is a major part of computational cancer genomics [3]. Another task is differentiating and characterizing them based on correlation with clinical phenotypes especially survival. Genetic mutations leading to low patient survival are in many senses the most important to study in terms of treating patients. Establishing the statistical significance of the correlation between two groups of patients, e.g. those carrying one of a set of mutations and those not has long been established. Thus characterizing the impact of known mutations on survival is well established. However there has been less work on using correlation with survival to find *de novo* mutations in the first place[2].

The difficulty in using correlation with survival (or any other statistic calculated over groups of patients) to identify collections of genes is size of the problem space. There are over six thousands of genes in the The Cancer Genome Atlas[4] leading to thousands of orders of magnitude too many ways to group them. To explore this vast space, local search must be used to restrict the focus to strongly correlated sets.

2 METHODS AND ANALYSIS

We wish to identify the set of genes most correlated with survival. Each set of genes divides patient into two groups, those carrying mutations in one or more of the genes in the set and those not carrying any mutation. Evaluating any set of genes is then as simple as evaluating any division of patients. This is done with the logrank test, as is standard for survival analysis[1]. It is not feasible to test all possible sets. It is also infeasible to deterministically find the set of maximal weight. However local search allows us to find the maximal set efficiently with high probability under certain assumptions about the underlying data.

2.1 *Model*

In our simulated model there are m patients $\mathcal{P}$ divided into two groups of samples P_1, correlated with lower survival, and P_2, correlated with higher survival. There are also n genes $\mathcal{U}$ divided into a survival correlated gene set G and its relative complement $\mathcal{U} - G$. Specifically for each sample p_i, the survival information consists of a survival time t_i and censoring information c_i. For p_i in P_1, the real survival time, s_i, and the censoring time, r_i, come from exponential distributions with parameters λ_1 and λ_c respectively; if $s_i \leq r_i$ then $t_i = s_i$ and $c_i = 1$, otherwise $t_i = r_i$ and c_i = 0. Survival data (time and censoring) for a sample p_i in P_2 are generated in a similar way, with s_i, and the censoring time, r_i, coming from exponential distributions with parameters λ_1 and λ_c respectively.

We view mutations as a $m \times n$ binary matrix A, where A_{ij} is 1 if sample i is mutated in gene j. The correlated set G only is mutated in P_1, while the rest of the genes are mutated randomly across all patients. For each random gene g there is a probability p_g that it is mutated in a sample in, independently of all other events. Each sample in P_1 has exactly one mutated gene g mutated in G chosen uniformly at random.

Our parameters were set to reflect real clinical data as much as possible, although this was not possible in setting the parameters of somatic mutations. $|P_1|$ was set to 15% of m. λ_1 was set to 1 and λ_2 was varied. λ_c was set to censor 20% of all samples in expectation (this censors more frequently P_2). $|G|$ was set to 5. The number and frequency of mutations of random genes were taken from real data sets.

2.2 *Logrank test*

The logrank test is only interested in the order events, not when they actually occur. To establish the order, we assume the rows of A are sorted so that $t_1 < t_2 < \ldots < t_m$. for a set $\mathcal{T} = \{g_1, g_2, \ldots, g_k\}$ of k genes we denote by $\mathcal{M}(\mathcal{T})$, the set of samples in which at least one of the genes in $\mathcal{T}$ is mutated: $\mathcal{M}(\mathcal{T}) = \left\{ i : \sum_{g_j \in \mathcal{T}} A_{ij} > 0 \right\}$ we denote by $\bar{\mathcal{M}}(\mathcal{T})$, the set of samples where no gene in $\mathcal{T}$ is mutated. For a set $\mathcal{T}$ of genes and any sample i, let $x_{\mathcal{T}}(i) = 1$ if $i \in \mathcal{M}(\mathcal{T})$, and $x_{\mathcal{T}}(i) = 0$ otherwise. For a sample i we define its weight as

$$w_i = c_i - \sum_{j=1}^{i} \frac{c_j}{m - j + 1}$$

For a set $\mathcal{T}$ of genes we define its weight as

$$\mathcal{W}(\mathcal{T}) = \frac{\sum_{i=1}^{m} x_{\mathcal{T}}(i) w_i}{\sqrt{\frac{|\mathcal{M}(\mathcal{T})|(m - |\mathcal{M}(\mathcal{T})|)}{m(m-1)} \left(\sum_{i=1}^{m} c_i \left(1 - \frac{1}{m-i+1} \right) \right)}}$$

This signed weight is maximized for sets correlated with low survival and minimized for sets correlated with high survival. It always has mean 0 and the denominator is the standard deviation of the numerator assuming $|\mathcal{M}|$ is a fixed ration of m. In the limit it thus goes to $\mathcal{N}(0, 1)$.

2.3 *Local search*

Using local search to explore the possible sets of genes requires a notion of adjacency, where adjacent sets of genes are probably close in weight. The most natural such notion is a directed acyclic graph where A gene set b is the child of gene set a if b is a with one additional gene. Thus starting with a root of the empty set one can consider larger and larger sets of genes. One could also consider deletions, but it is very unlikely that this would be helpful over random genes, and it removes the guarantees of the acyclic structure. At each set the search algorithm considers all sets containing the initial set and a single gene not in that set and keeps some number of them for future consideration. In the first greedy algorithm, steepest ascent hill climbing, we simply keep one gene. In our improved branching algorithm we take all sets above a threshold determined by the current set.

It is worth noting that while children are probabilistically bound to their parents in weight (especially under the model of random genes), there are very few deterministic guarantees about weight. Negative genes can combine into positive sets and vice-versa. This is because when considering the union of mutations across genes, Patients with few mutations are given the same weight as patients with many mutations in the set.

2.4 *Steepest ascent hill climbing*

The simplest form of local search is steepest ascent hill climbing (SAHC). This greedy algorithm simply stores a single set of genes at each step. It then uses this set to continue the process. It inherently finds sets of genes dominated by initial elements. Small sets found by SAHC being so dominated could potentially be found by examining individual genes, but under our union operator, it can find individually worse genes that fit together with initial strong genes to create better sets than simply combining highly weighted genes.

2.5 *Branching agorithm*

Since we are interested in sets of genes that are difficult to discover by looking at their genes individually, it is important that we hold on to less promising subsets, giving them the chance to build into the best sets. Thus in the branching algorithm (BA) instead of only selecting the best gene we select the k best genes and build out a tree of searched sets. Determining the appropriate value of k, or which genes to keep, is a difficult task, because it requires balancing false negatives, where we lose correlated sets with false positives, which bog down our runtime. The following argument establishes that as the number of samples grows, a very simple and generous thresholding policy is sufficient to guarantee with high probability, a low degree of branching, and thus a feasible runtime. More detailed analysis reveals that this asymptotic view is idealistic, and in the range of our computation, we will be forced to put a strict upper limit on k.

2.6 *Branching agorithm thresholding analysis*

Consider the following threshold to determine children of a given set in our search tree. Suppose we currently have a set of genes $\mathcal{T}_0$ and we are considering adding one gene $g \in \mathcal{U} - \mathcal{T}$ to $\mathcal{T}_0$ to create a child. We keep the set $\mathcal{T}_1 = \mathcal{T}_0 \cup \{g\}$ as a child if the weight of the set of samples mutated in $\mathcal{T}_1$, but not $\mathcal{T}_0$ is positive. That is $\sum_{j=1}^{m} x_{\mathcal{T}_1}(j)(1 - x_{\mathcal{T}_0}(j))w_j > 0$. Ignoring the numerator this is effectively when $\mathcal{W}(\mathcal{T}_1) > \mathcal{W}(\mathcal{T}_0)$.

Theorem: As long as $\mathcal{W}(\mathcal{T}_0) \in \omega(1)$, The probability of keeping $\mathcal{T}_1$ goes to 0 as m goes to ∞, if g is a random gene.

Proof: Assume the following. m is arbitrarily large. The data is uncensored i.e. $c_i = 0$ (this is possible if we throw out censored data, which only decreases $m$by a constant factor). $|\mathcal{M}(\mathcal{T}_0)| = k_0 = \phi m$. Let gene g be mutated in $k_1 = \psi m$ samples (chosen uniformly at random) outside of $\mathcal{M}(\mathcal{T}_0)$ (in expectation $\psi = (1 - \phi)p_i$).
Let $W_0 = \mathcal{W}(\mathcal{T}_0)$. We wish to choose k_1 distinct samples out of $\mathcal{P} - \mathcal{M}(\mathcal{T}_0)$. Let X_i be the weight of i^{th} sample selected. Let $\mu = \mathrm{E}[X_i]$, this will be the same for all X_i. Let $Y_0 = 0$ and $Y_i = X_i - \mu + Y_{i-1}$ for $i > 0$. Clearly Y_i is a martingale. Further since $X_i < 1$, $Y_i < 1 - \mu$. This let's us bound $P(Y_{k_1} > -k_1\mu) = P(\sum_{i=1}^{k_1} X_i > 0)$ with Azuma's bound and this exactly when we threshold.
Azuma's bound states that for martingale Y_i, where $Y_{i+1} - Y_i < c$ $P(Y_n - Y_0 > t) < e^{-t^2/2nc^2}$. In this case we have

$$
\begin{aligned}
P(\sum_{i=1}^{k_1} X_i > 0) &= P(Y_{k_1} > -k_1\mu) \qquad (1)\\
&< e^{-(k_1\mu)^2/2k_1(1+\mu)^2} \qquad (2)\\
&= e^{-\psi m(\mu^2/2(1+\mu)^2)} \qquad (3)
\end{aligned}
$$

That probability goes to 0 as long as $\mu^2 \in \omega(1/m)$

$$\begin{aligned}
\mu &= \mathrm{E}[X_i] && (4)\\
&= \frac{1}{m - |\mathcal{M}(\mathcal{T}_0)|} \sum_{i=1}^{m} (1 - x_{\mathcal{T}_0}(i)) w_i && (5)\\
&= \frac{1}{m - \phi m} \left(\sum_{i=1}^{m} \left(1 - \sum_{j=1}^{i} \frac{1}{m - j + 1} \right) - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i \right) && (6)\\
&= \frac{1}{(1-\phi)m} \left(\sum_{i=1}^{m} (1 - H_m + H_{m-i}) - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i \right) && (7)\\
&= \frac{1}{(1-\phi)m} \left(m - mH_m + \sum_{i=0}^{m-1} H_i - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i \right) && (8)\\
&\approx \frac{1}{(1-\phi)m} \left(m - m(\gamma + \log(m)) + \gamma + \sum_{i=1}^{m-1} (\gamma + \log(i)) - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i \right) && (9)\\
&= \frac{1}{(1-\phi)m} \left(m - m\log(m) + \log((m-1)!) - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i)) w_i \right) && (10)\\
&\approx \frac{1}{(1-\phi)m} \left(m - m\log(m) + (m-1)\log((m-1)) - (m-1) - \sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i \right) && (11)\\
&\approx \frac{1}{(1-\phi)m} \left(1 - \log(m) - \sum_{i=1}^{m} w_i x_{\mathcal{T}_0}(i) \right) && (12)\\
&< -\frac{\sum_{i=1}^{m} w_i x_{\mathcal{T}_0}(i)}{(1-\phi)m} && (13)
\end{aligned}$$

Where $H_i \approx \gamma + \log(i)$ is the i^{th} harmonic number, γ is the Euler-Mascheroni constant, and (11) follows from Sterling's approximation. Since,

$$\begin{aligned}
W_0 &= \mathcal{W}(\mathcal{T}_0) && (14)\\
&= \frac{\sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i}{\sqrt{\frac{|\mathcal{M}(\mathcal{T}_0)|(m - |\mathcal{M}(\mathcal{T}_0)|)}{m(m-1)} \left(\sum_{i=1}^{m} \left(1 - \frac{1}{m-i+1} \right) \right)}} && (15)\\
&= \frac{\sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i}{\sqrt{\frac{\phi m(m - \phi m)}{m(m-1)} (m - H_m)}} && (16)\\
&\approx \frac{\sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i}{\sqrt{\phi(1-\phi)m}} && (17)
\end{aligned}$$

we have $\sum_{i=1}^{m} x_{\mathcal{T}_0}(i) w_i = W_0 \sqrt{\phi(1-\phi)m}$ and

$$\mu < -W_0 \sqrt{\frac{\phi}{(1-\phi)m}} \qquad (18)$$

Thus $\mu^2 \in \omega(1/m)$, when $W_0 \in \omega(1)$. This concludes the proof.

The above analysis ignores ϕ and ψ as constants. They are however very small constants and can be almost as low as $1/m$, in both the real and simulated data. This is sufficient to drive the provably negligible probability of failing to threshold to over $1/2$, even for moderately large W_0, which allows for exponential blow up in the run time. This means that k needs be set to a maximum of a constant in the algorithm and the depth of the search must be limited to keep the algorithm feasible.

2.7 *Null Hypotheses*

While the logrank test is itself a measure of the significance of our results. It is a measure we are using in finding the results and further it makes underlying assumptions about the generation of the data. It is thus unfit to simply use logrank test to generate a p-value. Instead two permutation tests will be use. Each views a different permutation of the data as the null hypothesis for the generation of the mutation and survival data. The significance is given by the frequency of the algorithm finding a result of equal or greater weight. Thus if a particular dataset does follow the assumptions of logrank and returns consistently higher than expected weights, this will not affect the returned significances.

The first hypothesis H_0^P is the permutation of the survival data, with mutation matrix fixed. It selects a permutation of m elements uniformly at random and applies it to the survival times and censoring data jointly. The second hypothesis H_0^M is the permutation of each column of the mutation matrix. It independently selects a permutation of m elements uniformly at random for each column and applies them to each column. The survival data is unchanged. The H_0^P removes correlation between survival data and the mutation data, but preserves correlations between genes. H_0^M removes both correlations.

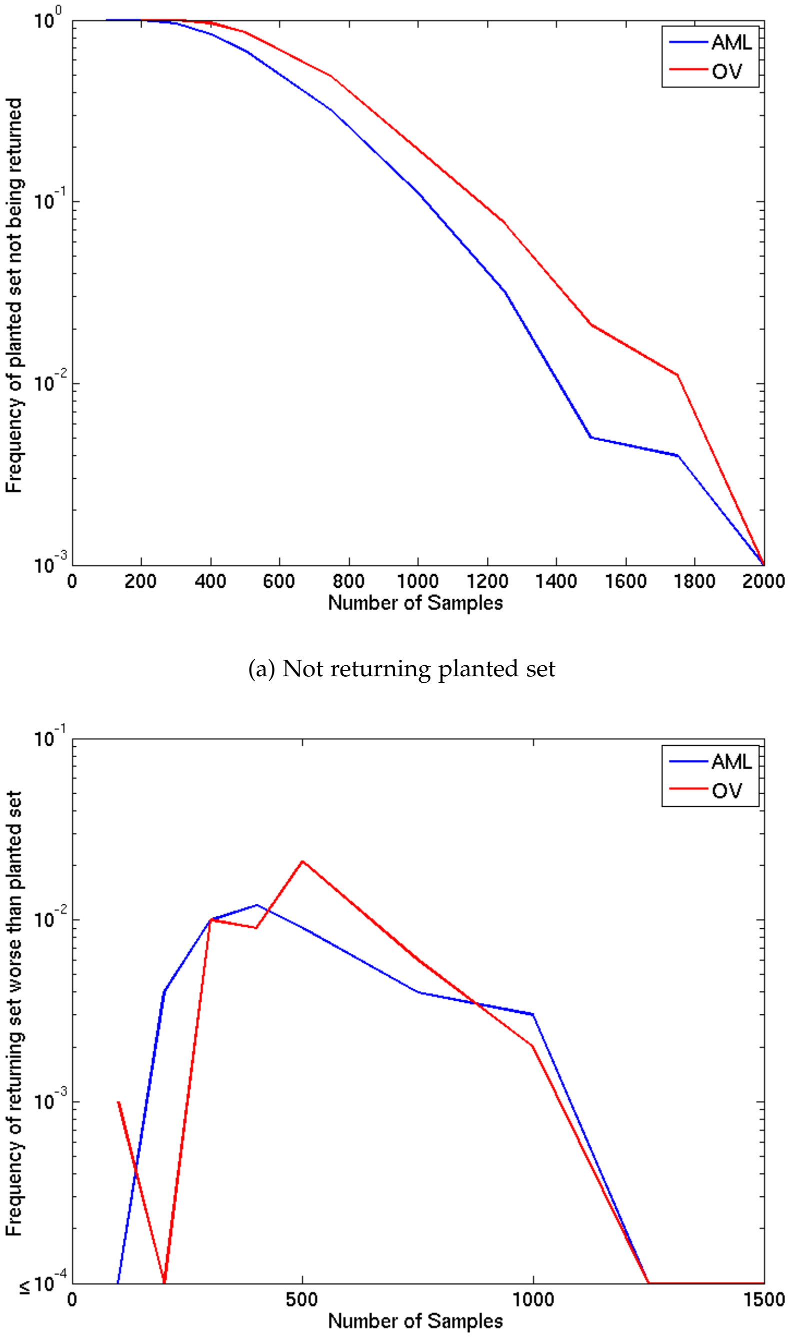
3 RESULTS

3.1 *Simulated Data*

Data was simulated according to the model described above for varying number of sampled patients m. A set G of 5 genes was planted and the remaining random genes were generated using frequencies from ovarian (OV) and acute myeloid leukemia (AML) datasets. The patients in P_1 were expected to live 20 months and the patients in P_2 were expected to live 60.

The first question assessed was how often will the algorithms return the planted set as the best set found. Both algorithms were run to a depth of 5 gene sets and the branching algorithm was bounded to keeping at most 4 sets at each level. Since th AML dataset was smaller than the OV set, it led to fewer random genes and fewer possibilities for a random set to do better than the planted set. This is shown in SAHC's performance in Figure 1a, where performance is better of AML. By the time there are 2,000 samples, the planted set is almost always returned in both datasets. Since the algorithm fails frequently at lower sample sizes, it is worth asking whether the algorithm fails to find the optimal set and whether the planted set is optimal. The second is more easily answered Figure 1b shows that the set returned by SAHC is almost always of weight at least that of the returned set.

This might make BA seem irrelevant if SAHC can already recapture all that is desired in the model. BA can still outperform SAHC by returning a ranked list of weighted sets (instead of just a single set). If the planted set is near the top of the list this can be viewed as a partial success. Figure 2 shows that BA returns the planted set in the top two significantly more often than as the top result, And in the top 10 results significantly more often. It is not shown, but under the parameterization used, BA very seldom returned the planted set at all if it was not in the top 10. There is also the question of BA and SAHC's performance when they do not return the planted set. BA dominates SAHC, since it returns any set SAHC would return, but that does not say how often that set is the top set. Figure 3 shows that BA initially far outperforms SAHC, but this diminishes as they both start returning the



(a) Not returning planted set

(b) Returning a set worse than planted set

Figure 1: The frequencies of different failures of SAHC

planted set, the advantage drops off as it must.

Figure 4 shows both algorithms *p*-values under all tests dropping off together. This shows how the significance of the planted set increases with the number of samples under both algorithms, which somewhat validates both the model and the algorithms.

3.2 *Real Data*

BA and SAHC were run on three datasets of glioblastoma multiforme (GBM), kidney renal clear cell carcinoma (KIRC), and AML. The sets returned were all of 5 genes, to match with simulated data, and because there was no cleaner break in logrank weight or *p*-values for smaller or larger sets. Every

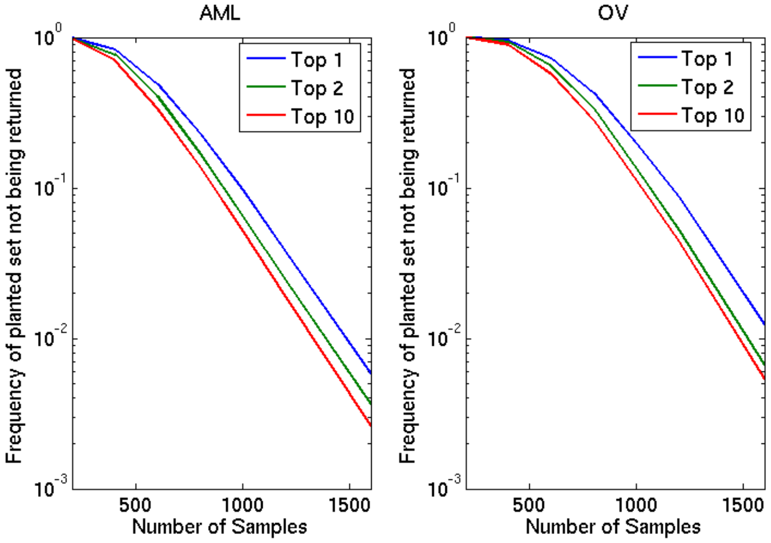


Figure 2: The frequencies of BA not returning the best set

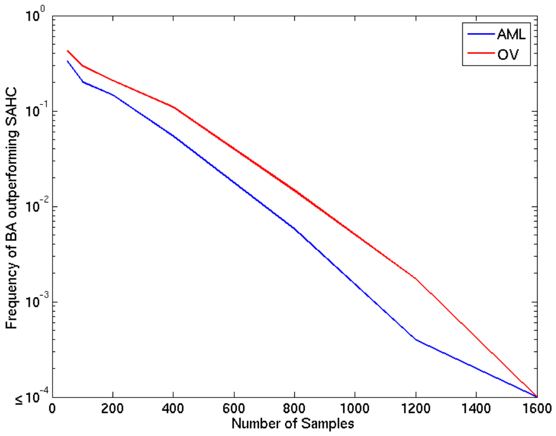


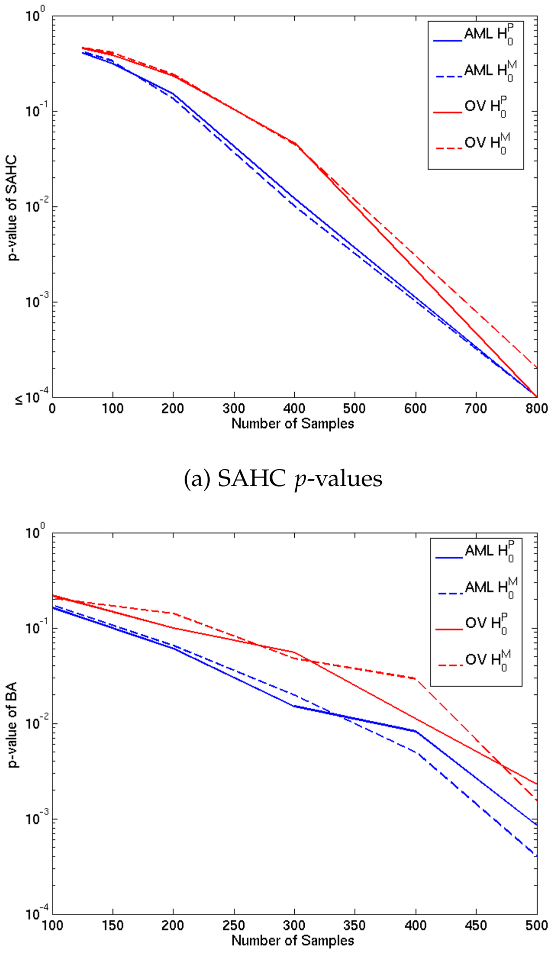
Figure 3: The frequencies of BA outperforming SAHC

top set found by BA, was the set returned by SAHC, again demonstrating SAHC's performance. p-values were calculated just using BA, since it is inherently stronger than SAHC. All gene sets are listed in order of the level to which they were added to BA. Thus the initial genes can be viewed as most significant.

The GBM dataset yielded the gene set {TBC1D2, CASKIN2, BID, SLC9A4, HDAC9}. It had logrank weight of 4.54, an H_0^P p-value of 0.9100, and an H_0^M p-value of 0.9775. The KIRC dataset yield the gene set {CDCA2, KIF27, TDRD7, PDE4C, OGT}. It had logrank weight of 7.30, an H_0^P p-value of 0.2733, and an H_0^M p-value of 0.3016. The AML dataset yield the gene set {TP53, DNMT3A, RUNX1, FAM5C, MLL-MLLT3}. It had logrank weight of 5.24, an H_0^P p-value of 0.0004, and an H_0^M p-value of 0.0006. GBM and KIRC datasets were significantly larger than AML. This could possibly explain AML's relative strength when it came to p-values. the AML dataset was dominated by a few strong genes, whereas the other sets had many middling genes, some of which became much stronger under permutation.

4 CONCLUSION

Using local search to find sets *de novo* somatic mutations strongly correlated with low survival is theoretically and empirically feasible. Using the current



(a) SAHC p-values

(b) BA p-values

Figure 4: The significance of the two algorithms

model of such collections, results should be significantly improve when datasets grow into the low thousands of samples. SAHC is a very strong preliminary technique for finding such sets, but BA is still more powerful, especially if the data does not necessarily follow the intended model.

Future work could generalize the model and explore other parameterizations specifically the mean survival of P_1 and P_2. Finding the limit of the use of the algorithms on the proximity of the two means could show how much impact a set needs on survival to be detected. Another important generalization are models with multiple planted sets, since the biological explanation of the model requires multiple such sets. Other local search algorithms could also be considered, but most are not viewed to be suitable by the author. A more serious theoretical treatment of the problem space should also be attempted. The alternative hypothesis (of a correlated set), deserves its own theoretical treatment. This would combine well.

5 ACKNOWLEDGEMENTS

I would like to thank Professors Eli Upfal and Fabio Vandin for advising me through, reading, and encouraging me to pursue my thesis.

[1] John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*. Wiley-Interscience, 2011.

[2] Fabio Vandin, Patrick Clay, Eli Upfal, and Benjamin J Raphael. Discovery of mutated subnetworks associated with clinical data in cancer. pages 55–66, 2012.

[3] Fabio Vandin, Eli Upfal, Benjamin J Raphael, et al. Finding driver pathways in cancer: models and algorithms. *Algorithms for Molecular Biology*, 7(1):23, 2012.

[4] RG Verhaak, Katherine A Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D Wilkerson, C Ryan Miller, Li Ding, Todd Golub, Jill P Mesirov, et al. Cancer genome atlas research network. integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. *Cancer Cell*, 17(1):98–110, 2010.