Discovering Combinatorial Interactions in Survival Data

David A. duVerle\textsuperscript{1,*}, Ichiro Takeuchi\textsuperscript{2}, Yuko Murakami-Tonami\textsuperscript{3,4}, Kenji Kadomatsu\textsuperscript{1} and Koji Tsuda\textsuperscript{1}

\textsuperscript{1}Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
\textsuperscript{2}Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan
\textsuperscript{3}Division of Molecular Oncology, Aichi Cancer Center, Nagoya, Japan
\textsuperscript{4}Department of Molecular Biology, Nagoya University Graduate School of Medicine, Nagoya, Japan

Associate Editor: Prof. Martin Bishop

ABSTRACT

Motivation: While several methods exist to relate high-dimensional gene expression data to various clinical phenotypes, finding combinations of features in such input remains a challenge, particularly when fitting complex statistical models such as those used for survival studies.

Results: Our proposed method builds on existing ‘regularisation path-following’ techniques to produce regression models that can extract arbitrarily complex patterns of input features (such as gene combinations) from large-scale data that relate to a known clinical outcome. Through the use of the data's structure and itemset mining techniques, we are able to avoid combinatorial complexity issues typically encountered with such methods and our algorithm performs in similar orders of duration as single-variable versions. Applied to data from various clinical studies of cancer patient survival time, our method was able to produce a number of promising gene-interaction candidates whose tumour-related roles appear confirmed by literature.

Availability: An R implementation of the algorithm described in this paper can be found at \url{https://github.com/david-duverle/regularisation-path-following}

Contact: dave.duverle@aist.go.jp

1 INTRODUCTION

From their inception, high-dimensional genomic data, such as obtained through genome-wide expression microarrays, have been used to identify genes that affects survival or tumour reoccurrence time spans among cancer patients (Van De Vijver et al., 2002; Bovelstad et al., 2007). Survival data generally contains partially-known observations (e.g., when clinical follow-up of the patient ends before a decisive event) requiring the use of regression models that can specifically handle censored data. Cox proportional hazards model (Cox, 1972) is one such model that combines advantages of both parametric and nonparametric approaches to statistical inference, making it ideally adapted to the type of data obtained in clinical trials.

Due to the high-dimensionality and small sample size of gene expression data, it is desirable to add a penalisation component in fitting the Cox model (Dudoit et al., 2002; Van De Vijver et al., 2002; Ghosh, 2003), with \( \ell_1 \)-norm often preferred for its ability to drive sparsity of the model and select a concise set of variables (gene expression values, mutation types, etc.) (Tibshirani et al., 1997; Gu and Li, 2005). Different methods have been suggested (Lin and Wei, 1989; Gu and Li, 2005; Park and Hastie, 2007) for fitting \( \ell_1 \)-penalised Cox model. Park and Hastie (2007), in particular, proposed a method to compute the regularisation path of \( \ell_1 \)-penalised Cox model: producing a series of Cox models that have different levels of complexity and sparsity.

As for many models in systems biology, it has been widely shown (Hanahan and Weinberg, 2000; Tibshirani et al., 2002) that the gene regulatory pathways of cancer involve non-linear gene interactions. While models based on linear combinations of gene expression may accurately approximate more complex interactions for some tasks, it can be desirable to specifically identify combinatorial covariates for such purpose as the identification of synthetic lethal genes (Kaelin, 2005). However, all current methods rely on the ability to enumerate potential input variables: while it is computationally feasible to examine each single gene in such a way (even for a large microarray), issues of exponential complexity quickly arise when considering interactions between more than one gene at a time.

In this paper, we extend the approach in Park and Hastie (2007) to handle combinatorial interactions among genes. We deal with issues of combinatorial explosion and computational complexity by taking advantage of itemset mining techniques (Uno et al., 2004). Using this approach, virtually limitless combinations of genes and phenotypes, grouped in itemsets of boolean variables, can be used as single predictor variables in the model. Our proposed algorithm computes the regularisation path of \( \ell_1 \)-penalised Cox models that account for the effects of combinatorial gene interactions on survival.

Beyond proportional hazards models, our itemset-based method can be applied to any regression model with convex loss, each time making use of the input’s structure and
sparcity to sidestep complexity issues, while at the same
time guaranteeing that events along the regularisation path
(values of the regularisation parameter for which a change
occurs in the model structure) are exhaustively explored.

In the rest of this paper, section 2 first outlines our
general approach for adapting existing path regularisation
techniques to work with patterns of discretised input features
instead of single continuous values. Section 3 details the
mathematical basis for our algorithm and illustrate its
application to proportional hazard models using Cox’s partial
likelihood as loss function (with further detailed proofs as
Supplementary Material). Finally, section 4 presents
qualitative and quantitative results obtained by applying our
method to different survival datasets.

2 APPROACH

2.1 $\ell_1$-Penalised Maximum Likelihood Estimation

A common defining feature to many major regression models,
such as Generalised Linear Models (GLM) or previously
mentioned Cox model, is the use of a loss function to fit the
parameters of otherwise analytically-intractable problems.
Adding an $\ell_1$ penalty term to the original loss criterion results
in the typical estimation problem:

$$\beta(\lambda) = \arg\min_\beta (-L(y; X\beta) + \lambda \|\beta\|_1)$$

where $L$ denotes the log-likelihood function with respect to
the given data $(X, y)$. $\beta$ is the vector of coefficients that
needs to be estimated and $\lambda$ the regularisation parameter.

For values of $\lambda$ tending toward infinity, all coefficients in $\beta$
will be forced to 0, whereas as $\lambda$ decreases, more coefficients
will have non-null values (that is, more predictor variables
will be used in the model estimation).

2.2 Regularisation Path-Following Algorithm

Among various methods for solving $\ell_1$-regularised problems
similar to (1), the use of so-called ‘regularisation path-
following’ algorithms (Hastie et al., 2005; Park and Hastie,
2007) is of particular interest for their ability to finely control
the number of active variables in the model, regardless of the
dimensionality of the input. The general idea behind path-
following is to study variations of the $\lambda$ parameter in the
space of $\beta$ coefficient values (see Figure 1): by decreasing
the value of $\lambda$, starting from the maximum $\lambda_{max}$ for which $\beta$
is non-null, we can find a sequence of all discrete values of $\lambda$,
for which new coefficients of $\beta$ changes between null and non-
null (corresponding to a particular predictor variable exiting
or entering the regression model). The resulting sequence of $\lambda_k$
and associated optimal $\beta(\lambda_k)$ allow us to model the data
at varying levels of sparsity.

Park and Hastie (2007) suggested a path-following
algorithm for $\ell_1$-regularised GLM that uses a predictor-
corrector approach to efficiently find all $\lambda_k$ and the
coefficients of the model associated with each level of
regularisation. If we define the ‘active set’, $A_{\lambda_k}$, as the
set of non-null indices in the coefficient vector $\beta(\lambda_k)$, their
algorithm can be defined as a loop over four main steps:

1. **Predict**: Starting with a known $\beta(\lambda_{k-1})$ and $\lambda_k$: the
next target value of $\lambda$, estimate $\beta(\lambda_k)$ using a piecewise
linear approximation of $\beta$, under the assumption that $A$
remains unchanged.

2. **Correct**: Solve the associated convex optimisation
problem to find the exact value of $\beta(\lambda_k)$ (using the linear
approximation as a warm start).

3. **Update active set**: By confronting the new values of $\beta$
to the optimality conditions of the problem, update
$A$ (i.e. add/remove predictors from the model). Repeat
step 3 if necessary to adjust $\beta$.

4. **Decrease $\lambda$**: Analytically find the exact value of $\lambda_{k+1}$,
at which the active set will next change.

It is worth noting that, when an $\ell_1$-regularised model is
fitted to high-dimensional small sample data, sparse models
are usually selected (based on some model selection criteria).
Therefore, we do not really have to compute the ‘entire’
regularisation path (from $\lambda_0$ to 0). The algorithm is usually
terminated for a value of $\lambda$ where the size of the active set $A$
is still much smaller than the input dimension.

Because steps 1 and 2 only use variables in the current
active set, $A$, they can be performed at little computing cost
for values of $\lambda$ where $|A|$ remains much smaller than the
number of variables. Steps 3 and 4 require solving simple
equations for each possible input variable (in linear time of
the input’s dimension).

In their work, Park and Hastie (2007) showed that, along
with GLM, their algorithm could also easily be applied to
the Cox proportional hazards model. In fact, it can be shown
that their results hold for any loss-based model fitting task,
provided a loss function that exhibits certain mathematical
properties (see section 3 and Supplementary Material).

2.3 Finding Combinatorial Covariates

When the linear model is extended to combinatorial interaction
terms, the input dimension increases exponentially
due to the combinatorial explosion of gene interactions.
Of the steps enumerated in section 2.2, the predictor and
corrector steps only deal with the small subset of covariates
currently in the active set $A$ and therefore do not need to
be changed. On the other hand, updating the active set in

![Fig. 1. Schematic representation of the regularisation path in the space of $\beta$. Successive values of $\beta(\lambda_k)$ can be approximated using $\frac{\partial \beta(\lambda)}{\partial \lambda}$.](http://bioinformatics.oxfordjournals.org/Downloaded from)
step 3 and finding the next value of $\lambda$ at which an update event will occur in step 4, both potentially require examining a number of feature combinations that grows exponentially with the order of the interactions considered.

One practical approach to dealing with issues of combinatorial explosion and computational complexities in steps 3 and 4, is to take advantage of the input’s structure in order to efficiently explore its space. By discretising our input (gene expressions or other clinical data) and considering all possible sets of such binary variables, we can use itemset mining techniques (Uno et al., 2004; Saigo et al., 2007) to preserve the computational efficiency of the path-following algorithm despite a high dimensional input.

We show that step 3 can be reduced to a weighted itemset mining problem, easily solvable using existing optimisation techniques (see Methods section 3.1.3), while step 4 requires solving a particular form of fractional programming problem, for which we developed an efficient pruning approach (see Methods section 3.1.4). Our method can therefore overcome those computational complexity issues, and identify complex interactions (between two or more factors) that contribute to the response model, at varying degrees of sparsity (controlled by the penalisation component).

2.4 Application to Cox Proportional Hazards Model

We applied our modified version of the path-following algorithm to the Cox Proportional Hazards (Cox PH) model, where patient survival (or any timed event) is used as a response, allowing for missing data due to right-censorship. To estimate this model, we seek to maximise a so-called Log Partial Likelihood function (see Methods section 3.2) for a given set of data. As predictors, we use discretised values of the response model, at varying degrees of sparsity (controlled by the penalisation component).

3 METHODS

In this section we give a quick overview of the path-following algorithm first presented by Park and Hastie (2007) and the necessary changes to work on combinatorial interactions:

3.1 Path-following algorithm

Let $J(\beta)$ be the criterion from (1):

$$J(\beta) := -L(y;X\beta) + \lambda \|\beta\|_1 \quad (2)$$

In the regularisation path, we consider the optimal parameter vector $\beta$ as a function of the regularisation parameter $\lambda$, and represent the optimal parameter vector at $\lambda$ as $\beta(\lambda)$. We can write the optimality condition as follows:

$$H(\beta(\lambda), \lambda) := \frac{\partial J(\beta)}{\partial \beta} \bigg|_{\beta(\lambda)} = 0 \quad (3)$$

Our goal is to compute the path of solutions of (3) for all the $\lambda$. If we only consider the range of $\lambda$ where the active set $A$ does not change (noting $\beta_A$: the restriction of $\beta$ to the active set $A$), the partial change of the optimality condition (3) with respect to $\lambda$ must satisfy:

$$\frac{\partial H(\beta(\lambda), \lambda)}{\partial \lambda} = \frac{\partial H}{\partial \lambda} + \frac{\partial H}{\partial \beta_A} \frac{\partial \beta_A}{\partial \lambda} = 0 \quad (4)$$

3.1.1 Predictor step

In each predictor step, we assume that the current active set, $A$, does not change. In the $k$-th predictor step, we use a linear approximation to predict $\beta$ with the current active set:

$$\hat{\beta}_A(\lambda_{k+1}) := \beta_A(\lambda_k) + (\lambda_{k+1} - \lambda_k) \frac{\partial \beta_A(\lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} \quad (5)$$

3.1.2 Corrector step

We also assume that the active set $A$ does not change during each corrector step. Any convex optimisation algorithm can be used to minimise the penalised loss function (2). The use of $\hat{\beta}_A(\lambda_{k+1})$ as an initial starting point ensures that an optimal solution can be found in a small number of iterations.

3.1.3 Active set update

After each corrector step, it is necessary to identify all new features that should enter $A$. If we consider the set $P$ of all possible patterns, up to a given length, of binarised input features (e.g.: “gene A over-expressed and gene B under-expressed”) and assign each such pattern an index value, for any $\ell \in \{1, \ldots, |P|\}$, we note $x_\ell \in \mathbb{B}^n$ (where $n$ is the total number of observations) the indicator vector for the matching pattern. Our goal is to identify such values of $\ell$ that contribute to minimise the loss function (2), and for which the matching value of the parameter vector $\beta$ should be non-null (noted as $\beta_\ell$ being ‘active’ and $\ell$ being in the ‘active set’ $A$).

With the feature notation $X := \{x_1, \ldots, x_\ell\}$, we define:

$$w_i := \frac{\partial L}{\partial \beta} x_i \quad \ell := \sum_{i=1} w_i x_i \quad (6)$$

Assuming strong complementarity slackness, we obtain the following result (see Supplementary Material for detailed proof):

**Theorem 1.**

$$\beta_\ell \text{ is active } \iff |x_\ell| = \lambda \quad (7)$$

Therefore, if $|x_\ell| \geq \lambda_{k+1}$ after the corrector step, $\ell$ (and its associated parameter $\beta_\ell$) must be added to the active set $A$.

If $\ell$ were an easily enumerable feature (such as in the case of single gene expression level), it would be computationally feasible to exhaustively enumerate all values of $c_\ell$ for all possible $\ell$. In our case, however, $\ell$ can match an arbitrarily long pattern drawn from the power set of all binarised features: the number of such features grows exponentially with the maximum size of the patterns, making the problem highly impractical for sets of more than 2 or 3 items. However, as long as $c_\ell$ can be rewritten as linear sums of $x_{\ell, i}$, finding all such $\ell$ can be accomplished in reasonable time, using frequent itemset enumeration techniques.

Since the values $w_i$ in the linear sum defined in (6) do not depend on $\ell$ (and are constant for $\lambda \in [\lambda_{k+1}, \lambda_k]$), finding all items $c_\ell \geq \lambda_{k+1}$ is equivalent to finding all itemsets with weighted support above $\lambda_{k+1}$ (the symmetrical problem of also finding $\{c_\ell \ | -c_\ell \geq \lambda_{k+1}\}$ is then trivial). To solve this problem, we use the LCM program\(^1\) (Uno et al., 2004), which provides an exhaustive enumeration of frequent itemsets in guaranteed polynomial time per itemset.

If any variable is added to the active set $A$, or removed (indices $\ell \in A \beta_\ell = 0$), we go back to the corrector step (where the new values of $c_\ell$ are first re-computed). These two steps are repeated until the active set does not change, thus guaranteeing that the solutions are optimal.

3.1.4 Step length

To determine the optimal step length (the minimal value by which the regularisation parameter must be decreased in order for

---

\(^1\) [Link to the mentioned website](http://research.nii.ac.jp/~uno/codes.htm)
the active set to change), we need to solve a similar problem, this time involving the ratio of two separate frequent itemset mining optimisation problems.

If we define the step length:

\[ \Delta \lambda_k = \lambda_{k+1} - \lambda_k \]

the minimum decrement of \( \lambda \) for which the active set \( A \) changes (a variable is added or removed), it can be shown (see Supplementary Material for detailed proof) that:

**Theorem 2.**

\[ \Delta \lambda_k = \min_{\lambda \in A} \left\{ \frac{\lambda_k - c^+_{d_t}}{d_t - 1} \lambda_k + c^+_{d_t}, \Delta_{\text{non-active}}, \lambda_k \right\} \]

where \( \min^{+} \) is the smallest strictly positive value, \( d_t := \frac{\partial \lambda}{\partial x} \) and \( \Delta_{\text{non-active}} \) is obtained by:

\[ \Delta_{\text{non-active}} = \min_{\lambda \in A} \left\{ -\beta_1^+ \left( \frac{\partial \beta_1}{\partial \lambda} \big|_{\lambda=\lambda_k} \right)^{-1} \right\} \quad (8) \]

We note that \( \Delta_{\text{non-active}} \) only depends on the variables in the active set and can be easily computed. On the other hand, much like in section 3.1.3, exhaustively computing the values of the first two expressions in (8) for all \( \ell \in A \) is not computationally feasible given the dimension of our input.

We designed an exploratory approach using bounds on each subproblem to efficiently prune the search tree and drastically reduce the number of solutions explored.

First, we observe that both expressions can be rewritten as optimisation problems of the form:

\[ \min_{\lambda \in A} \left\{ \frac{\lambda_k - c^+_{d_t}}{d_t - 1} \lambda_k + c^+_{d_t}, \Delta_{\text{non-active}}, \lambda_k \right\} \quad (9) \]

where \( \forall \ v : p_i, q_i \in R \) only depend on the variables in the active set \( A \) (and can therefore be easily computed) and \( \kappa_p, \kappa_q : \text{constant terms} \) defined as:

\[ \phi^*_v = \min_{\{v_t, \eta \} \in \mathbb{R}^n} \left\{ \kappa_p + \sum_{i} p_i x_{i t} + \sum_{j} q_i x_{j t} \right\} \quad (10) \]

where \( n \) is the number of non-zero values for the itemset \( \ell \) being considered. \( \{p_i\}, \{q_i\} \in \mathbb{R}^n \).

While the general form of this problem is shown to be NP-hard (by association to the well-known NP-complete subset sum decision problem), it has an easy polynomial solution (Hammer et al., 1968; Boros and Hammer, 2002) if certain conditions hold.

With the following notation, separating positive and negative terms in the sums of \( p_i \) and \( q_i \):

\[ \forall i, p_i = p_i^+ - p_i^- + p_i^2, p_i^+ > 0; p_i^- := \sum p_i^- x_{i t}; \quad p_i := \sum p_i x_{i t} \]

\[ \forall i, q_i = q_i^+ - q_i^- + q_i^2, q_i^+ > 0; q_i^- := \sum q_i^- x_{j t}; \quad q_i := \sum q_i x_{j t} \]

we have the following result:

**Theorem 3.** For a given itemset \( \ell \), it is not necessary to explore any superset of \( \ell \) if either of the following conditions holds:

\[ (\kappa_q - q^*_\ell \geq 0) \land (\phi^*_v \geq \text{curmin}) \]

\[ (\kappa_q + q^*_\ell \leq 0) \land (\phi^*_v \geq \text{curmin}) \]

where \( \text{curmin} \) is the current minimum value found by the algorithm up until itemset \( \ell \).

A much faster (\( O(1) \)), albeit slightly weaker, pruning condition can also be obtained (see proof in Supplementary Material):

**Theorem 4.** For a given itemset \( \ell \), it is not necessary to explore any supersets of \( \ell \) if either of the following conditions holds:

\[ (\kappa_q - q^*_\ell \geq 0) \land \left[ \frac{\kappa_p - p^*_\ell}{\kappa_q + q^*_\ell} \geq \text{curmin} \right] \lor \left[ \frac{\kappa_p - p^*_\ell}{\kappa_q + q^*_\ell} \leq 0 \right] \]

\[ (\kappa_q + q^*_\ell \leq 0) \land \left[ \frac{\kappa_p + p^*_\ell}{\kappa_q + q^*_\ell} \geq \text{curmin} \right] \lor \left[ \frac{\kappa_p + p^*_\ell}{\kappa_q + q^*_\ell} \leq 0 \right] \]

While this pruning-based method loses some of its efficiency as the regularisation parameter \( \lambda \) decreases and the model becomes less sparse, for the range of values of \( \lambda_k \) treated, it remains well within the reach of standard computing equipment (under a minute on a single 3.2 GHz CPU core).

### 3.2 Application to Cox Proportional Hazards Model

In order to demonstrate the potential of our model, we applied it to the Cox model. This model uses survival data of the general form \( \{(x_i, y_i, \delta_i)\}_{i=1}^n \), where \( x_i \in \mathbb{R}^d \) is the vector of risk factors: for instance gene expression levels. In practice the \( x_i \) used by our method are vector of binary indicators of under- or over-expression (possibly in combination): \( y_i > 0 \) is the time observed (survival until an event or censoring); \( \delta_i \in \{0, 1\} \) is a binary variable indicating whether an event has taken place \( (\delta_i = 1) \) or the observation was right-censored \( (\delta_i = 0) \).

The Cox regression model (Cox, 1972) for the hazard of death at time \( t \) can be expressed as:

\[ h(t) = h_0(t) \exp(\beta^T X) \quad (11) \]

where \( h_0(t) \) is the baseline hazard function, \( \beta \in \mathbb{R}^d \) is the vector of parameters and \( X = \{X_1, \ldots, X_d\} \) is the vector of risk factor variables with corresponding sample value of \( x_i \) for the \( i \)-th sample.

However, it is not necessary to know \( h_0(t) \) in order to infer the regression parameters, thanks to the use of the Log Partial Likelihood function of the Cox model (Tibshirani et al., 1997), defined as:

\[ \mathcal{L}(\beta) = \sum_{i, \delta_i = 1} \left( \beta^T x_i - \log \left( \sum_{j, \delta_j \geq \delta_i} \exp(\beta^T x_j) \right) \right) \quad (12) \]

Refer to the Supplementary Material for the exact computation of the criterion \( c_\ell \) (6) in the case of the Cox Proportional model.

### 3.3 Gathering Synthetic Candidates

In order to extract as many interaction candidates as possible, while avoiding the risk of overfitting the data, we repeatedly run the path-following algorithm on a randomly chosen subset of the input. It has been shown (Meinshausen and Bühlmann, 2010) that the use of such sampling method with regularised methods of variable selection provides a good estimator of the original data.

On each run of the algorithm, we keep feature combinations that show a significantly improved predictive power over the linear models (likelihood ratio test \( p\)-value < 0.01). We aggregate all such combinations and rank them by Kaplan-Meier test \( p\)-value in order to produce a list of candidate interactions positively or negatively affecting the timed outcome.

As could be expected, a few combinations will tend to reoccur multiple times across successive iterations of the algorithm, while a large number only occurs once or twice. We hypothesised and verified a posteriori (see Supplementary Material) that combinations with low number of occurrences might be overfitting a particular iteration’s training subset and have poor
generalisation power. We therefore set an additional screening thresholds on the list of interactions: keeping only those that occur in at least 4 (out of 100) iterations. This threshold value was selected as giving the best compromise between ratio of false positives and overall number of interactions found (see details in Supplementary Material).

Independent testing shows remarkable stability of the list of selected interactions for a large-enough number of iterations. With our chosen occurrence and p-value thresholds, the final list of variables sees very little change after approximately 50 iterations (see plot in Supplementary Material). This trend is also confirmed when using an independent test: none of the rarely occurring combinations added in later iterations turn out to be significant in the test subset. For our experiment, we therefore set the total number of total iterations to 100: a value that once again seems to offer a good compromise between exhaustivity and the risk of false discovery.

4 EVALUATION

4.1 Datasets

To test our method, we used two datasets publicly available: survival studies of neuroblastoma (Oberthuer et al., 2006) and breast cancer (Van De Vijver et al., 2002) patients. In both studies, cDNA microarray assays of gene expression (10,163 probes for 9,878 unique genes and 24,158 probes for 23,031 unique genes, respectively), along with (right-censored) survival data, were available for n = 251 and n = 295 patients, respectively. In both cases, after setting aside a test subset (25% of all instances), the algorithm was iteratively applied on randomised subsets of the training data (95%) in a method similar to the leave-one-out procedure (Kearns and Ron, 1999).

For each study, gene expression data was normalised across arrays using standard methods (Yang and Thorne, 2003), then discretised in two binary classes depending on their distance to the mean (μ) using a threshold proportional to the standard deviation (σ): genes that are over-expressed (expression value above μ + θσ, where θ is a thresholding parameter, set to 1.5 in this instance) or under-expressed (below μ − θσ).

To compare the higher-order interactions found by our method to a linear combination search, we ran the original Park and Hastie (2007) algorithm on the same training datasets and ranked the resulting variables found by the order in which they entered the regularised model. These ranks appear in the result tables under the column ‘single-variable rank’ (‘NA’ indicates a variable that did not appear in any of the models fitted by the single-variable version of the algorithm before one of its default termination conditions were reached).

4.2 Analysis of Breast Cancer Data

The list of interactions found for Van De Vijver et al. (2002) (see table 1) not only features a large number of genes strongly associated with breast cancer prognosis in the medical literature, such as SLC2A3 (Sternlicht et al., 2006), CA9 (Span et al., 2003), RAB6B (van’t Veer et al., 2002), BBC3 (Cobleigh et al., 2005) or KIAA0882 (Abba et al., 2005), many of which do not appear at all in single-variable model fits (see single-variable ranks), it also features interesting examples of synthetic interactions: for example, the Kaplan-Meier plot for the interaction between BBC3 and KIAA0882 (Figure 2) shows perfect prediction of survival of all test samples (p < 0.0003), compared to the much less significant plot for BBC3 alone (p = 0.03), while a strong synthetic effect can be observed with BBC3 over-expressed (logrank p-value: 0.008, see plots in Supplementary Material).

Despite the overall small number of samples and difficulties to obtain good generalisation power from such small training and test subsets, these results hold fairly well in test. Logrank p-values computed over an independent test subset for all selected combinations show 6 out of 9 (66.7%) to be significant (p < 0.05), with 4 combinations (44%) still significant after Bonferroni correction for multiple-hypotheses testing.

4.3 Analysis of Neuroblastoma Data

The even smaller number of samples for Oberthuer et al. (2006) makes it difficult to obtain good generalised results (Table 2), however, the single interaction validated on the test subset (out of four interactions in total selected by our algorithm) not only shows strong predicting power on both subsets, but also involves two sequences strongly tied to breast cancer in literature. Locus BC046178 is associated with CENPW (previously known as C6orf173 or CUG2), a well-studied oncogene associated with apoptotic behaviours in tumour cells (Lee et al., 2007, 2010). Probe Hs458148 is a match for multiple genes including RPL10: a ribosomal protein-coding gene which has been found to be over-expressed in breast cancer tumours (Nagai et al., 2004). Although Hs458148 could also match other genes, its expression values in this dataset are highly correlated (Pearson’s coefficient: 0.63) with two other probes exclusively matching RPL10.

4.4 Model Validity and Computation Time

Although our goal is primarily not to create a predictor, but to gather input feature combinations (with promising synthetic lethality properties, in the case of cancer studies), we could still confirm that the model estimates produced...
by our method were sound and consistent with previous methods. Separating the original dataset in a training (75%), model-selection (12.5%) and test (12.5%) subsets and running nested cross-validation (100 iterations at the training level, each evaluated over 100 partitioning of the model-selection and evaluation subsets), we were able to compare the average log partial likelihood for both our algorithm and that of Park and Hastie (2007) (who use a $\ell_1$-penalised path-following algorithm that only selects single variables, hereafter referred to as single-variable algorithm or single-variable model), both on the test subset.

Using the breast cancer survival data from Van De Vijver et al. (2002), our algorithm gave a mean log-partial likelihood of $-121.00$ (sd: 27.56) compared to $-117.10$ (sd: 26.85) for the single-variable algorithm by Park and Hastie (2007), both significantly ($p < 2.2e-16$) higher than the Null model ($-123.28$, sd: 27.88), where no variables are used. With both algorithms, a large variance in the cross-validated results and overall middling performances are to be expected due to the small sizes of training, model-selection and testing subsets along with the typically high level of noise in microarray data. However, as the validation of the results in section 4.2 shows, there is still enough signal to detect very meaningful covariates.

Additionally, we ran our algorithm on a randomised version of the breast cancer data, where survival data had been shuffled so as to no longer match its particular gene expression data. Using the same experimental set-up as described in 4.1, the algorithm produced only two significant interactions ($p < 0.05$): one of which only occurred once (and therefore would not be selected under normal conditions), while the other, with a $p$-value of 0.03, was no longer significant after Bonferroni correction (correction factor: 36) for multiple-hypotheses testing. This is to be contrasted with the multiple Bonferroni-significant interactions found in regular data (see section 4.2).

Computing time, although consistently longer for our algorithm was still within reasonable distance of the single-variable version: with similar termination conditions and the same input data, a single run of our path-following algorithm took on average less than 5 minutes (281 s ± 83 s) on a quad-core 3.2 GHz CPU, compared to a little under a minute for Park and Hastie (2007) (36 s ± 6 s).

### Table 1. Interaction results for Van De Vijver et al. (2002)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>up.SLC2A3 * up.CA9</td>
<td>0.00153</td>
<td>0.000175</td>
<td>65</td>
<td>0.003432472</td>
<td>NA NA</td>
</tr>
<tr>
<td>dn.Contig56307 * up.RAB6B</td>
<td>0.00168</td>
<td>0.000392</td>
<td>15</td>
<td>0.097443906</td>
<td>NA NA</td>
</tr>
<tr>
<td>dn.BCC3 * dn.KIAA0882</td>
<td>5.21e-05</td>
<td>0.00043</td>
<td>22</td>
<td>0.0002761196</td>
<td>NA NA</td>
</tr>
<tr>
<td>up.KIAA0964 * up.SLC2A3</td>
<td>0.000254</td>
<td>0.00132</td>
<td>23</td>
<td>0.04875811</td>
<td>NA NA</td>
</tr>
<tr>
<td>up.GADD153 * up.SLC31A1</td>
<td>0.0147</td>
<td>0.0022</td>
<td>5</td>
<td>0.25696054</td>
<td>540 NA</td>
</tr>
<tr>
<td>dn.Contig41887_RC * dn.KIAA0252</td>
<td>0.0151</td>
<td>0.00387</td>
<td>13</td>
<td>0.01261452</td>
<td>NA NA</td>
</tr>
<tr>
<td>up.RAD51C * up.TIMELESS</td>
<td>0.0367</td>
<td>0.0168</td>
<td>4</td>
<td>0.001651706</td>
<td>NA NA</td>
</tr>
<tr>
<td>up.TGFBI * up.IGTA5</td>
<td>0.0195</td>
<td>0.0298</td>
<td>11</td>
<td>0.2221538</td>
<td>NA NA</td>
</tr>
<tr>
<td>dn.Contig41887_RC * up.UGT8</td>
<td>0.000172</td>
<td>0.0329</td>
<td>51</td>
<td>0.003777276</td>
<td>NA NA</td>
</tr>
</tbody>
</table>

Selected feature combinations, ranked by Kaplan-Meier $p$-value. Bonferroni-significant Kaplan-Meier test $p$-values are in bold (correction factor: $m = 71$). Total variables found with single-variable model: 585.

### Table 2. Interaction results for Oberthuer et al. (2006)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>up.BC046178 * up.Hs458148.20</td>
<td>0.0131</td>
<td>2.16e-07</td>
<td>36</td>
<td>0.01003018</td>
<td>NA 67</td>
</tr>
<tr>
<td>dn.THC1529413 * up.Hs172998.2</td>
<td>0.0199</td>
<td>0.00142</td>
<td>20</td>
<td>0.3228081</td>
<td>NA NA</td>
</tr>
<tr>
<td>dn.L233919 * up.USP1</td>
<td>0.0164</td>
<td>0.00561</td>
<td>61</td>
<td>0.2413684</td>
<td>89 NA</td>
</tr>
<tr>
<td>up.U92981 * up.SLC14A2</td>
<td>0.0147</td>
<td>0.0369</td>
<td>9</td>
<td>0.1264266</td>
<td>67 NA</td>
</tr>
</tbody>
</table>

Selected feature combinations, ranked by Kaplan-Meier $p$-value. Bonferroni-significant Kaplan-Meier test $p$-values are in bold ($m = 48$). Total variables found with single-variable model: 474.

5 CONCLUSION

In this paper, we presented an algorithm to follow the regularisation path of any $\ell_1$-regularised linear model fitting, using combinatorial interactions as covariates. While the path-following method has been applied to microarray data in the past (Park and Hastie, 2007), it was until now only able to deal with single-valued features, ignoring possible higher-order effect of gene interactions.

Our method makes uses of existing frequent-itemset mining techniques and novel imports from fractional programming to avoid the intractability issues of combinatorial input and produce a regression model of accuracy and run time comparable to the linear case. By running multiple iterations of the algorithm on subsampled datasets, we can produce...
ordered list of candidate interactions with strong predicting power.

The interactions found by applying our method to cancer study survival data include many genes that could not be found through linear models, yet show up in literature as strongly tied to these conditions, confirming the crucial importance of taking interaction effects into account in order to detect some of the weaker signal in gene expression data. While most significant interactions found by our method on experimental data were limited to two or three genes, there are no theoretical limitations to the size of interactions that can be searched, at no particularly higher computational cost, setting this method apart from other recent work on penalised selection of interactions in high-dimensional data (Bien et al., 2012).

The strong noise inherent to gene expression microarray likely prevents the detection of weaker signals between more than three genes, making it an attractive prospect to work with less noisy types of data where larger interactions might be detectable. In the future, we plan to extend our field of application to a wider range of biomedical data, such as the identification of SNP interactions (Schwender and Ickstadt, 2008), as well as leverage our model’s ability to deal with heterogeneous input, for example by including a wide range of clinical data in addition to the large-scale numeric data.

ACKNOWLEDGEMENT

Funding: This work was supported in part by a Grant-in-Aid for JSPS Fellows (No. 24-02709 to DdV); by MEXT KAKENHI (No. 23700165 to IT); by a Grant-in-Aid for Scientific Research on Innovative Areas (No. 23110002 to K.K.) from the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of Japan; by a Grant-in-Aid (No. 20390092 to K.K.; 24590376 to Y.M.-T.) from MEXT; and by funds from the Global COE program, MEXT, Science, and Technology (MEXT) of Japan; by a Grant-in-Aid for JSPS Fellows (No. 24-02709 to DdV); by MEXT Funding for Research on Innovative Areas (No. 23110002 to T.K.); and by a Grant-in-Aid for Scientific Research (C) (No. 24670041 to DdV).

REFERENCES


more important than genetic variation in regulating pai1 expression. Cancer Epidemiology Biomarkers & Prevention, 15(11), 2107–2114.


