Non-parametric survival analysis in breast cancer using clinical and genomic markers

Technical University of Denmark

Author:
Søren Sønderby
SN: 112391

Supervisors:
Ole Winther

March 31, 2014

Contents

List of Figures

List of Tables

Abstract

Background: New survival models based on Gaussian Processes (GP) and Random Forests (RF) have been developed, and have shown good performance in large cancer cohorts.
Purpose: To investigate if these new survival models can improve prediction of 10 year recurrence in a pooled dataset of breast cancer patients.
Data Sources: Breast cancer patients collected by (Haibe-Kains et al. 2012)
Data Extraction: Patient clinical data and gene expression data from several platforms were extracted. Clinical data, including receptor status, was incomplete. Methods for inference of ER, HER2 and PgR receptor status from gene expression data was developed. These methods work independenty of the gene expression platform. Recurrence predictors where extracted from expression data.
Results: A pilot study showed that RF survival had worse performance than GP based models. RF survival was not investigated further. Area under curve (AUC) scores for recurrence prediction in breast cancer patients was calculated for the models Cox GP model (CoxGP) and Cox proportional hazard (CoxPH). When appropriate, models were evaluated on dataset with different number of covariates.
Limitations: The included data is a pooled dataset and may be skewed.
Conclusion: CoxGP models show better performance than CoxPH. It is shown that addition of features extracted from gene expression data improve prediction of 10 year recurrence in both CoxGP and CoxPH models.

Chapter 1

Introduction

This project evaluates methods for survival analysis in breast cancer patients. Breast cancer is one of the most prevalent cancer types with more than 200,000 incidences and nearly 40,000 deaths per year in USA 2013. The incidence rate is the highest among cancer types in American women, as shown in figure 1.1 (Desantis et al. 2013). Risk stratification of breast cancer patients has traditionally relied on measurement of clinical markers such as tumor size, histological grading, age etc. combined with pathological markers such as presence or absence of estrogen receptor (ER), human epidermal growth factor receptor 2 (HER2) and progesterone receptor (PgR) in tumor tissue samples[1] (Goldhirsch et al. 2013; Galea et al. 1992).

In 2001 Sørlie et al. 2001 published a micro array based method for classifying breast cancers into intrinsic subtypes named luminal A, luminal B, HER2-enriched, and basal-like. These subtypes has been shown to add prognostic value for prediction in breast cancer patients (Parker et al. 2009). In 2002 Veer et al. 2002 published a 70-gene signature which classify patients as having good or bad recurrence risk. Since then a large number of gene signatures for prognostication of breast cancer patients has been published. The most prominent include OncotypeDX (Paik et al. 2004) and MammaPrint (Veer et al. 2002; Vijver et al. 2002). OncotypeDX predicts the risk of distant recurrence after 10 years in node negative, estrogen receptor positive patients. The prediction is based on a 21-gene signature.

[1] In this text all of the above mentioned features are called *clinical* features to distinguish them from features based on *genomic* data

Estimated New Cases*

Males			Females		
Prostate	238,590	28%	Breast	232,340	29%
Lung & bronchus	118,080	14%	Lung & bronchus	110,110	14%
Colorectum	73,680	9%	Colorectum	69,140	9%
Urinary bladder	54,610	6%	Uterine corpus	49,560	6%
Melanoma of the skin	45,060	5%	Thyroid	45,310	6%
Kidney & renal pelvis	40,430	5%	Non-Hodgkin lymphoma	32,140	4%
Non-Hodgkin lymphoma	37,600	4%	Melanoma of the skin	31,630	4%
Oral cavity & pharynx	29,620	3%	Kidney & renal pelvis	24,720	3%
Leukemia	27,880	3%	Pancreas	22,480	3%
Pancreas	22,740	3%	Ovary	22,240	3%
All Sites	**854,790**	**100%**	**All Sites**	**805,500**	**100%**

Estimated Deaths

Males			Females		
Lung & bronchus	87,260	28%	Lung & bronchus	72,220	26%
Prostate	29,720	10%	Breast	39,620	14%
Colorectum	26,300	9%	Colorectum	24,530	9%
Pancreas	19,480	6%	Pancreas	18,980	7%
Liver & intrahepatic bile duct	14,890	5%	Ovary	14,030	5%
Leukemia	13,660	4%	Leukemia	10,060	4%
Esophagus	12,220	4%	Non-Hodgkin lymphoma	8,430	3%
Urinary bladder	10,820	4%	Uterine corpus	8,190	3%
Non-Hodgkin lymphoma	10,590	3%	Liver & intrahepatic bile duct	6,780	2%
Kidney & renal pelvis	8,780	3%	Brain & other nervous system	6,150	2%
All Sites	**306,920**	**100%**	**All Sites**	**273,430**	**100%**

Figure 1.1: The ten leading cancer types by gender in USA 2013 (Desantis et al. 2013)

MammaPrint predicts 5-year metastatic recurrence risk as good or bad by using the expression of 70 genes (Veer et al. 2002; Vijver et al. 2002). A review of these and other methods is available in Marchionni et al. 2008.

This project investigates how clinical and genomic features can be combined for survival analysis in breast cancer patients. Typically survival has been modeled using the Cox proportional hazard model (CoxPH), but this model comes with rather strong assumptions. Recently several new models based on Gaussian processes (GP) and random forests (RF) have been developed (Joensuu et al. 2012; Ishwaran et al. 2008). A recent paper by Joensuu et al. 2012 successfully uses GP based survival models to model recurrence risk

in gastrointestinal stromal tumor patients. We use the same approach for breast cancer patients. The new models are compared to classical models and evaluated by their ability to model survival in breast cancer patients.

1.1 Project Proposal

Micro array data and clinical data of breast cancer patients will be collected from public available databases. The recurrence free survival time of breast cancer patients is modeled using the methods introduced by Joensuu et al. 2012 and Ishwaran et al. 2008, these new models will be compared to CoxPH, Nottingham prognostic index (NPI) and St. Gallen consensus criteria (STG), the latter two being risk stratification schemes for breast cancer patients (Goldhirsch et al. 2013; Galea et al. 1992).

A baseline model will be established by modeling recurrence using clinical markers, i.e. age, histological grade, tumor size, treatment status. These models will then be expanded with tumor receptor status and lastly we will investigate the effect of adding features derived from micro arrays.

Several problems need to be resolved during the project. Several of the datasets do not include tumor receptor status. It is possible to infer the receptor status from micro array data, but current methods rely on specific Affymetrix probes[2] which are not available in a pooled dataset. The problem is further complicated by the fact that data is included from studies using several different platforms. Current methods for receptor inference are typically developed using a single platform and the performance using other platforms is not known. We develop methods for inference of receptors status that are independent of the micro array platform.

A number of different features, derived from micro array data, have been published. These generally suffer from the same problems as mentioned above. It needs to be investigated whether current features can be generalized to other platforms than the ones they where developed on.

[2]Affymetrix is the producer of one of the major platforms for micro array measurements.

1.2 Project Outline

The following is a brief outline of this report and the project in general. The project commenced with data collection. To limit the complexity of the project data was collected from a patient repository created by Haibe-Kains et al. 2012 which includes most breast cancer data with micro array data up to 2012. A pilot study additionally included data from the METABRIC study (Curtis et al. 2012). The METABRIC data was not included in the final evaluation because we did not gain access to the micro array data until late in the project. The included data is described in section 2.9.

The used survival models, GP based, random survival forest, Cox proportional hazard, Nottingham prognostic index and St. Gallen consensus criteria are described in section 2.1. Gaussian process based and Random survival forest (RSF) methods, are computationally intensive, especially combined with cross validation, feature selection and optimization of hyper parameters. A pilot study was used to determine which survival model that should be evaluated. The pilot study showed that the GP based models generally performed better or on par with RSF based models. To limit the computational requirements further investigations where therefore limited to GP based survival models. A detailed description of the pilot study is given in section 2.10.

To investigate the effect of different features the survival models were evaluated using different datasets:

- A baseline model including only clinical data (tumor size, histology, age, treatment status, nodal involvement)
- a model which additionally adds tumor receptor status (ER, HER2, PgR)
- a model which additionally adds features derived from micro array data, e.g. prediction of survival time from micro array data.

The different datasets are described in section 2.11 and the included covariates are presented in section 2.9.

A method called top scoring pairs (TSP) was used for inference of tumor

receptor status, the method is described in section 2.7.2. This section also describes previously developed methods for receptor inference, which are based on Gaussian mixture models. These mixture based models generally performed poorly on the data used in this project, probably because they where developed on a single platform and the data in this study comes from several platforms. The TSP method alleviate some of the problems by being a gene expression rank based methods which makes it invariant to (most) scaling and normalization problems. The results of receptor inference are presented in 3.1.

Inclusion of features derived from micro array data proved to be more difficult than expected. The main problem was that currently published features are based on specific Affymetrix probes which are difficult to map to Entrez gene ID's. Micro array derived features are described in section 2.8.

The included models are described in section 2.1 to section 2.5, and the performance of the evaluated models is presented in section 3.3. The report concludes with a discussion of the obtained results in chapter 4.

The project strives to be fully reproducible. Except for the pilot study, the code for reproducing the results, figures and tables is provided. A detailed description of the code is given in section 2.12.

Chapter 2

Methods

2.1 Survival Models

The following sections briefly explain survival analysis, the format of survival data, the likelihood function and methods for modeling survival times. Survival analysis is presented in section 2.1.1. Survival data is often given as right-censored data, this data type is described in section 2.1.2. The likelihood function, used for fitting parameters in a statistical model, is presented in section 2.1.3. Section 2.2 explains the CoxPH model which is a commonly used survival model (Cox 1972). Section 2.3 describes survival models based on GP's and section 2.4 describes survival models based on random forests. Lastly NPI and STG models are described in section 2.5.

2.1.1 Survival Analysis

Survival analysis is the analysis of time-to-event data. We assume that the survival time T is a random variable representing the survival time. T is defined in the interval $[0, \infty)$. Let $f(t)$ be the probability density function for T, see figure 2.1 (left panel). $F(t)$ is the cumulative distribution function of $f(t)$, defined in equation 2.1. $F(t)$ gives the probability that T is less than or equal to t.

In survival analysis we are typically interested in the probability that an event did not happen before t, this is called the survival function and is defined in equation 2.2. The survival function gives the probability that

T is strictly greather than t. Figure 2.1 (middle panel) shows a survival function. $S(t)$ is a monotone decreasing function and $S(0) = 1$ and $S(\infty) = 0$.

The hazard function, $h(t)$, is defined in equation 2.3. It is defined as the instantaneous rate of occurrence at time t, given that no event occurred for the individual up to time t. Figure 2.1 (right panel) shows an example of a hazard function. $h(t)$ is not a probability because we divide the numerator by Δt and $h(t)$ may then be larger than 1 which is not allowed for probabilities.

$$F(t) = P(T \leq t) = \int_0^t f(u)du \tag{2.1}$$

$$S(t) = 1 - F(t) = P(T > t) = \int_t^\infty f(u)du \tag{2.2}$$

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = \frac{f(t)}{S(t)} \tag{2.3}$$

In the remaining part of this section we will derive some relationships between $f(t)$, $F(t)$, $S(t)$ and $h(t)$, which are mathematically equivalent descriptions of the random variable T. First we derive the last equality in equation 2.3. In equation 2.3 we use the product rule of probability[1] to rewrite the equation to:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t, T > t)}{P(T > t)\Delta t}. \tag{2.4}$$

From equation 2.2 we have that the denominator can be rewritten as $S(t)\Delta t$. The numerator can be simplified by noting that $t \leq T < t + \Delta t$ is a subset of $T > t$, see figure 2.2. We can then simplify the numerator to $P(t \leq T < t + \Delta t)$. Rewriting equation 2.4 we get

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t)}{\Delta t} \frac{1}{S(t)}. \tag{2.5}$$

The numerator can now be rewritten into two the difference between the probabilities $P(T \leq t + \Delta t)$ and $P(t \leq T)$, which corresponds to the cumulative distribution functions $F(t + \Delta t)$ and $F(t)$. Inserting this into equation

[1] $p(Y|X) = \frac{p(Y,X)}{p(X)}$

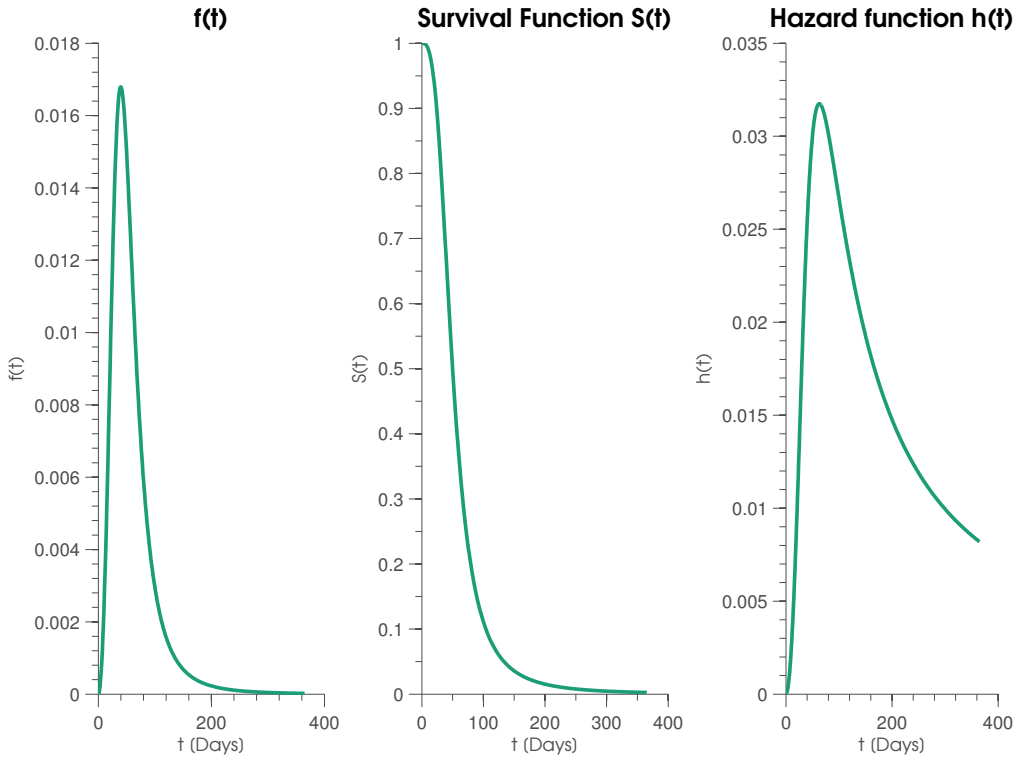

Figure 2.1: *Left panel:* Example probability density function for T, *middle panel:* Survival function, *right panel:* Hazard function.

2.5 the first fraction on the right hand side becomes a derivative which can be solved[2]

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{P(T \leq t + \Delta t) - P(t \leq T)}{\Delta t} \frac{1}{S(t)} \\ &= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \frac{1}{S(t)} = \frac{f(t)}{S(t)}. \end{aligned} \tag{2.6}$$

Equation 2.6 proves that the relationship given in equation 2.3 is correct. Equation 2.7 show the relationship between $f(t)$ and $S(t)$:

$$f(t) = -\frac{d}{dt} S(t) = -\frac{d}{dt}[1 - F(t)] = -[-f(t)] = f(t) \tag{2.7}$$

[2] $f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$

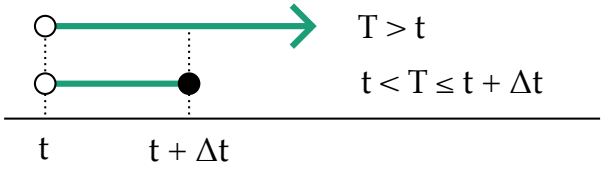

Figure 2.2: Shows that $t \leq T < t + \Delta t$ is a subset of $T > t$. A filled dot includes the point in the set, a not filled dot does not include the point in the set.

The relationship between $S(t)$ and $h(t)$ is given in equation 2.8. A relationship between $f(t)$ and $h(t)$ can be found by integrating booth sides of equation 2.8 to get equation 2.9.

$$h(t) = -\frac{d}{dt}\ln[S(t)] \tag{2.8}$$

$$h(t) = -\frac{d}{dt}\ln[S(t)] \Leftrightarrow -\int_0^t h(u)du = \ln[S(t)] \Leftrightarrow \tag{2.9}$$

$$S(t) = \exp\left[-\int_0^t h(u)du\right] \Leftrightarrow f(t) = h(t)\exp\left[-\int_0^t h(u)du\right]$$

We can prove equation 2.8 by using the derivative of $\ln$ (eq. 2.10) and the chain rule of differentiation (eq. 2.11). In equation 2.12 we use equation 2.11 with $S(t)$ substituted for u and insert the derivative of $\ln(S(t))$ from equation 2.10. Equation 2.12 can then be solved to $\frac{f(t)}{S(t)}$ which by equation 2.3 is equal to $h(t)$.

$$\frac{d}{du}\ln(u) = \frac{1}{u} \tag{2.10}$$

$$\frac{d}{dt}\ln(u) = \frac{d\ln(u)}{du}\frac{du}{dt} = \frac{1}{u}\frac{du}{dt} \tag{2.11}$$

$$h(t) = -\frac{d}{dt}\ln(S(t)) = -\frac{1}{S(t)}\frac{dS(t)}{dt} = \frac{f(t)}{S(t)} = h(t) \tag{2.12}$$

2.1.2 Right-censored Data

For right-censored data we do not know the exact outcome for all samples, but only that the survival time exceeds some value c or that an event have

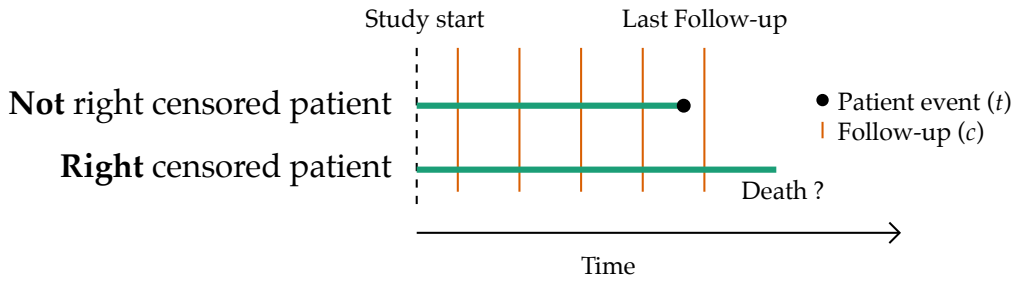

Figure 2.3: Explanation of right-censored data. For the right-censored patient $t > c$ and we do not know the exact survival time for the patient. For the not right-censored patient $t \leq c$ and the exact survival time is known.

occurred (e.g. relapse, death, failure etc.). A patient is right-censored if the study ends before the patient has an event. Figure 2.3 explains right-censoring. In the figure the *not* right-censored patient dies before the last follow-up and the exact survival time is known. For the right-censored patient we do only know that her survival time exceed the last follow-up but the exact survival time is unknown. Let c_i be the last follow-up for patient i and t_i be the time of death for patient i. Patient i is right-censored if $t_i > c_i$, if $t_i \leq c_i$ the patient is not right-censored.

2.1.3 Likelihood in Survival Analysis

Likelihood functions are used for fitting the parameters in statistical models. The likelihood function is a function of the parameters in the statistical model it is defined as

$$L(\beta|X, y) = P(y|\beta, X). \tag{2.13}$$

X is the covariates and y is the observed outcomes. The likelihood function is typically viewed as function of the parameters β[3]. We now define the likelihood function for survival models. For survival analysis using right-censored data two cases exists: *a)* the patient has an event before censoring in which case we know the exact survival time of the patient. In case *b)* the patient does not have an event before censoring, in which case we only

[3]A more indebt description of likelihood functions is available at: `http://cs229.stanford.edu/notes/cs229-notes1.pdf`

know that the survival time exceeds the censoring time, but we do not the exact time to event. Before the likelihood is defined note that equation 2.3 can be rewritten as:

$$h(t) = \frac{f(t)}{S(t)} \Leftrightarrow f(t) = h(t)S(t) \tag{2.14}$$

In case a we use equation 2.13 and equation 2.14 to write the likelihood contribution of individual i. Individual i has survival time t_i which is the probability density at time t_i. Using this we write the likelihood as

$$L_i = f(t_i) = S(t_i)f(t_i) \qquad \text{(Not censored)}. \tag{2.15}$$

In case b we know that the survival time exceeds the censoring time, which is equal to $S(t)$ by definition. The likelihood contribution is then

$$L_i = S(t_i) \qquad \text{(Censored)}. \tag{2.16}$$

We combine equation 2.15 and equation 2.16 into a single expression by introducing a right-censoring indicator δ. δ_i is 1 if sample i is right-censored and 0 otherwise. Using the censoring indicator we can combine equation 2.15 and equation 2.16 into a likelihood for a single sample as shown in equation 2.17 and for all n samples as shown in equation 2.18.

$$L_i = h(t_i)^{1-\delta_i} S(t_i) \tag{2.17}$$

$$L = \prod_{i=1}^{n} L_i = \prod_{i=1}^{n} h(t_i)^{1-\delta_i} S(t_i) \tag{2.18}$$

2.2 Cox Proportional Hazard Model

The CoxPH model is a model that handles right-censored data. We will not go into detail about the CoxPH models, but only specify the model and briefly discuss the proportional hazard assumption. An in depth description of CoxPH models can be found in Cox 1972. For a single individual i the CoxPH model is defined as:

$$h(t|x_i) = h_0(t) \cdot \exp(x_i \cdot \beta) \tag{2.19}$$

The CoxPH models gives the hazard at time t for patient i with covariates x_i. β is the regression coefficients of the model. h_0 is the baseline hazard

function, which is equal to the hazard function for patients with covariates all 0. Using the relationships between $h(t)$, $S(t)$ etc. we can get the survival function from the hazard defined in equation 2.19. The CoxPH model assumes that the baseline hazard is independent of the covariates, and the exponential part is independent of t. In the basic CoxPH model all covariates have multiplicative effect and no interaction occurs.

We now explain the "proportional hazard" assumption in the CoxPH model. First we define the hazard ratio (HR) between the individual or groups i and j as:

$$\text{HR} = \frac{h(t|x_i)}{h(t|x_j)} = \frac{h_0(t) \cdot \exp(x_i \cdot \beta)}{h_0(t) \cdot \exp(x_j \cdot \beta)} = \exp(x_i - x_j)\beta. \tag{2.20}$$

The CoxPH model assumes that the hazard ratio between groups is constant over time, i.e. the hazard ratio is independent of time. This follows from the right hand side of equation 2.20 which does only depends on the difference in covariates between group i and j. We can reformulate equation 2.20 as

$$\theta = \frac{h(t|x_i)}{h(t|x_j)} \Leftrightarrow h(t, x_i) = \theta \cdot h(t, x_j), \tag{2.21}$$

where θ is equal to the hazard ratio, which is constant over time between any two groups. Equation 2.21 shows that the hazard function for any group can be calculated as some constant, θ, times the hazard function for another group, i.e. the hazard functions between groups are proportional. The proportional hazard assumption is not met if the hazards between groups change over time. An example of this is surgical intervention where the intervention group have high hazard after surgery but low long time hazard. For the non-intervention group the short term risk is low but may rise with time.

The parameters in the CoxPH models, β, can be found by partial likelihood optimization, partial because the likelihood does only explicitly considered non-censored samples. Refer to Cox 1972 or Ibrahim, Chen, and Sinha 2001 for an indebt description of partial likelihood for CoxPH models. The likelihood can be written as

$$PL(\beta|D) = \prod_{i=1}^{n} \left[\frac{\exp(x_i'\beta)}{\sum_{l \in \mathcal{R}_i} \exp(x_l\beta)} \right]^{1-\delta_i}, \tag{2.22}$$

n is the number of samples and δ_i is 1 if patient i is right censored and 0 otherwise (Ibrahim, Chen, and Sinha 2001, p. 16).

2.3 Gaussian Process Based Survival Models

The GP based survival models used here where introduced in Joensuu et al. 2012 and are implemented in the MATLAB package `gpstuff` (Vanhatalo et al. 2013). The section assumes basic familiarity with GP processes. A short description of GP's are available in the appendix 5.3, p. 86 for a comprehensive description refer to Rasmussen and Williams 2006 available at: `http://www.gaussianprocess.org/gpml/chapters/RW.pdf`.

The CoxPH model in equation 2.19 is extended by replacing the linear predictor $x_i\beta$ with $\eta_i(x_i)$, η being a GP. The extended model for the hazard rate for sample i is

$$h_i(t) = \exp(\log(h_0(t)) + \eta_i(x_i)). \tag{2.23}$$

The GP over $\boldsymbol{\eta} = (\eta_1, ...\eta_n)^T$ is defined as

$$p(\boldsymbol{\eta}|X) = \mathcal{N}(\mathbf{0}, C(X, X)), \tag{2.24}$$

where X is the matrix of covariates for all n samples and C is a covariance function which will be defined later. We further assume that the baseline hazard is piecewise constant as shown in figure 2.4. The hazard function is divided into K equal length intervals with cut points: $0 = s_0 < s_1 ... < s_K$ where $s_k > y_i \forall i = 1, ..., n$. Using the piecewise linear assumption the baseline hazard can be written as shown in equation 2.25 and equation 2.26 which is $\ln$ of the former.

$$h_0(t) = \lambda_k \qquad for\ t \in (s_{k-1}, s_k] \tag{2.25}$$

$$f_k = \ln(\lambda_k)\ for\ t \in (s_{k-1}, s_k] \tag{2.26}$$

A second GP is placed on $\mathbf{f} = (f_1, ..., f_K)^T$ and equation 2.23 can be written as

$$h_i(t) = \exp(f_k + \eta_i(x_i)) \qquad , t \in (s_{k-1}, s_k]. \tag{2.27}$$

The GP over $\mathbf{f}$ has the form

$$p(\mathbf{f}|\boldsymbol{\tau}) = \mathcal{N}(\mathbf{0}, C_\tau(\boldsymbol{\tau}, \boldsymbol{\tau})), \tag{2.28}$$

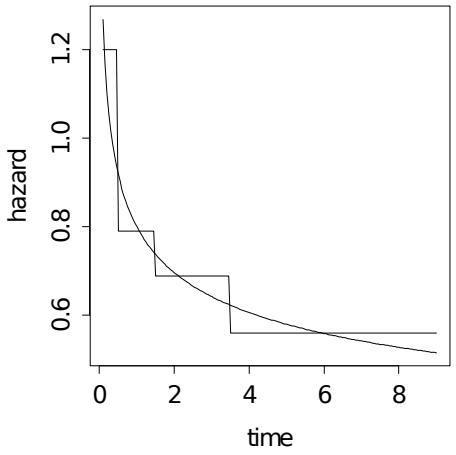

Figure 2.4: Hazard function with piecewise constant approximation overlaid.

here $\boldsymbol{\tau} = (\tau_1, ...\tau_K)$ is the vector of mean values of the K intervals that the hazard function is divided into and C_τ is the covariance function for $\mathbf{f}$.

The likelihood for equation 2.27 can be found by substituting the hazard rate into equation 2.17 and remembering the relationship between $h(t)$ and $S(t)$ given in equation 2.9. This then gives the likelihood of a single sample as

$$L_i = h_i(t_i)^{1-\delta_i} \exp\left(-\int_0^{t_i} h_i(u)du\right). \tag{2.29}$$

Substituting the definition of the Cox GP hazard function into equation 2.29 the likelihood of sample i can be written as

$$L_i = \left[\lambda_k \exp(\eta_i)\right]^{1-\delta_i} \exp\left(-[(t_i - s_{k-1})\lambda_k + \sum_{g=1}^{k-1}(s_g - s_{g-1}\lambda_g)]\exp(\eta_i)\right). \tag{2.30}$$

2.3.1 Covariance Functions

A GP is an interpolator, i.e. for data points that are similar we make similar predictions. In a GP we define similar via the covariance function.

A covariance function is a function that takes two data points as input and outputs the similarity. A popular covariance function is the squared exponential defined as

$$k(x_i, x_j) = \sigma^2_{\text{exp}} \left(-\frac{1}{2} \sum_{k=1}^{d} \frac{(x_{i,k} - x_{j,k})^2}{l_k^2} \right). \tag{2.31}$$

Here x_i and x_j is the vector of covariates for sample i and sample j. The length of x_i and x_j is d. In the squared exponential covariance function $\sigma 2_{\text{exp}}$ and l are considered hyper parameters. The task of learning in a GP is to tune the hyper parameters and choose an appropriate covariance function. Changing the hyper parameters will give the function different characteristics. l_k, the length scale in dimension k, determines the correlation scale in this dimension. A small value for l will make the predictions dependent on nearby points whereas a larger value will put more weight on far away data points. $\sigma 2_{\text{exp}}$ is determines the overall variability of the GP[4].

In this project a number of different covariance function were explored, among others squared exponential, Matern, Linear and Neural Network kernels. Experiments with different combinations of the above mentioned covariance functions where also performed. The choice of kernel function generally had minor impact on the performance of the GP models, but a combination of neural network kernel and constant kernel consistently showed good performance. Neural Network combined with constant kernel was used for booth the baseline hazard, $\mathbf{f}$, and the latent predictors (η) in the GP model.

2.3.1.1 Constant Kernel

Constant covariance kernel:

$$k_{\text{CON}}(x_i, x_j) = \sigma \tag{2.32}$$

[4] `http://skaae.shinyapps.io/test_project/` lets you play with the hyper parameters. The example is also available at `https://github.com/skaae/GP_shiny/` with instruction on how to run the example locally.

2.3.1.2 Neural Network Kernel

The neural network covariance function has the form:

$$K_{\mathrm{NN}}(x_i, x_j) = \frac{2}{\pi} \sin^{-1}\left(\frac{2\tilde{x_i}^T \Sigma \tilde{x}_j}{\sqrt{(2\tilde{x}_i^T \Sigma \tilde{x}_j) \cdot (2\tilde{x}_i^T \Sigma \tilde{x}_j)}}. \right) \tag{2.33}$$

$\tilde{x}$ is the vector of covariates augmented with 1, i.e. $(1, x_1, ..., x_d)$ and Σ is $\mathrm{diag}(\sigma)$

2.3.2 GP Tuning of Hyper Parameters

A GP model has a varying number of hyper parameters depending on the choice of covariance function. These hyper-parameters needs to optimized. Hyper parameters were optimized by maximizing the marginal likelihood. The marginal likelihood is the probability that the models assigns to the correct target given the covariates, i.e. $p(\mathbf{y}|X)$. For non-Gaussian observation models the marginal likelihood was optimized using Laplace approximation (Rasmussen and Williams 2006).

2.4 Random Survival Forest

Random survival forests are survival models based on random forests (Ishwaran et al. 2008; Breiman 2001). This section assumes that the reader has basic familiarity with random forests. Section 2.4.1 to section 2.4.3 describes Random Survival forest, the hazard function and tuning of hyper parameters.

2.4.1 Random Survival Forests

When a RSF is grown the following procedure is used:

1. Draw B bootstrap samples from the original data. Note that each bootstrap sample excludes on average 37% of the data, called out-of-bag data (OOB data).
2. Grow a survival tree for each bootstrap sample. At each node of the tree, randomly select p, candidate variables. The node is

split using the candidate variable that maximizes survival difference between daughter nodes.

3. Grow the tree to full size under the constraint that a terminal node should have no less than $d_0 > 0$ unique deaths.
4. Calculate a Cumulative Hazard Function (CHF) for each tree. Average to obtain the ensemble cumulative hazard function.
5. Using OOB data, calculate prediction error for the ensemble CHF. (Ishwaran et al. 2008)[5]

For clarification a *bootstrap* dataset of size n is constructed by drawing n samples **with** replacement from the original dataset. Candidate variables, p, mentioned in point 2, are covariates to be considered for splitting at each node in the tree. When the tree is grown, each node will split the dataset such that its daughters will have the largest possible difference in survival time. That is at each node we push dissimilar cases apart. The tree is grown until each terminal node has at least d_0 unique deaths. That is if we drop the bootstrap data (training data) down the tree each terminal node will have at least d_0 unique deaths. Figure 2.5 shows an example of a single tree.

Hyper parameters in the RSF is candidate features at each split, p, the number of unique deaths in each node, d_0, the number of trees in the forest. Lastly the number of splitting points to be considered for each candidate variable is a hyper parameter, i.e. when we try to push dissimilar cases apart how many threshold values should we try for each of the considered covariates.

2.4.2 Cumulative Hazard Function

For each terminal node a cumulative Hazard Function (CHF) is predicted. All cases in a specific terminal node share CHF. The CHF estimate ($\hat{H}_h(t)$)

[5]The description is a quote from (Ishwaran et al. 2008)

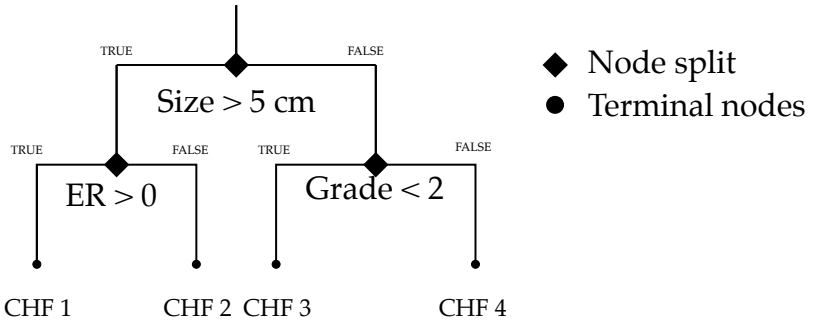

Figure 2.5: An example tree constructed from a bootstrap dataset. At each node split p covariates are considered for pushing samples apart. For each considered covariate the number of considered thresholds values, e.g. tumor size $>$ *5 cm*, is a hyper parameter. The tree is grown such that when all samples are dropped down the tree at least d_0 unique samples end in each terminal node. For each terminal node a Cumulative Hazard Function (CHF) is constructed.

at terminal node h in a single tree is:

$$\hat{H}_h = \sum_{l=t_{1,h}}^{t_{N(h),h}} \frac{d_{l,h}}{Y_{l,h}} \tag{2.34}$$

$Y_{l,h}$: Number of individuals at risk at $t_{l,h}$

$d_{l,h}$: Number of deaths at $t_{l,h}$

$t_{1,h} < t_{2,h}, ..., t_{N(h),h}$ are the distinct event times at terminal node h. To get the CHF estimate of individual i, with covariates x_i, drop x_i through the tree. The individual will end in some terminal node h. h's CHF is the estimated CHF.

As in random forest an ensemble of trees is trained, each using a different bootstrap dataset and different candidate features at the node split. Equation 2.34 is the CHF estimate for a single tree. For the ensemble we can either estimate the OOB CHF or the bootstrap CHF. H_{OOB} is the average over all predicted CHF's for which individual i is OOB, as shown in equation 2.35. The bootstrap estimate of the CHF is simply the average

CHF over all B trees, shown in equation 2.36.

$$H_{\text{OOB}} = \frac{\sum_{b=1}^{B} I_{i,b} H_b^*}{\sum_{b=1}^{B} I_{i,b}} \tag{2.35}$$

$$H_{\text{BOOTSTRAP}} = \frac{1}{B} \sum_{b=1}^{B} H_b^* \tag{2.36}$$

H_b^* : CHF from bootstrap tree b

B : Number of bootstrap trees

$I_{i,b}$: 1 if individual i is OOB for tree b else 0

Empirical evaluation of Random survival forest by Ishwaran et al. 2008 suggests that the method is insensitive to noise variables e.g. features with no information. This makes the method a good candidate for trying different genetic measures as predictors of survival.

2.4.3 RF Tuning of Hyper Parameters

The R package `RandomSurvialSRC` was used to grow the forest (Ishwaran and Kogalur 2013). The forest consisted of 1000 trees. The minimum terminal node size, number of candidate features at splits and the number of splitting points for each features were considered hyper parameters. Minimum terminal node size, candidate features at splits and number of splitting points where optimized using grid search, the values $1, 3, 5, 10$, $1, 3, 5, 10, 50$ and $0, 1, 3, 10$[6] were searched respectively. OOB error rate was used as criteria for selecting the best model.

2.5 Nottingham Prognostic Index and St. Gallen

Nottingham Prognostic Index (NPI) and St. Gallen consensus criteria (STG) are guidelines for stratification of breast cancer patients (Galea et al. 1992; Goldhirsch et al. 2003). NPI is described in section 2.5.1 and STG in section 2.5.2. NPI and STG are both designed to evaluate patient survival and not recurrence risk. Comparison of NPI and STG to models that

[6] 0 means all possible split are tried, see help for `rfsrc` package

are specifically trained for prediction of recurrence is therefore unfair. Performance of STG and NPI was used because no other widely used models where identified in the literature.

2.5.1 Nottingham Prognostic Index

The Nottingham prognostic index (NPI) predicts 15 years survival in breast cancer patients of age up to 70 at time of diagnosis. The NPI is defined in equation 2.37. Using equation 2.37 the NPI score is calculated which can then be translated to a risk group by using table 2.1.

$$\begin{aligned} \text{NPI} \quad = \quad & [\text{Size (cm)}] \times 0.2 + \\ & [\text{Nodes (lymph nodes, 1-3 by level)}] + \\ & [\text{Grade (1-3: good, moderate, poor)}] \end{aligned} \tag{2.37}$$

Risk	Score
Good	<3.4
Intermediate	[3.4-5.4]
Poor	>5.4

Table 2.1: NPI score classification (Galea et al. 1992)

.

2.5.2 St. Gallen Consensus

STG is based on the covariates tumor size, histological grade, nodal involvement, her2 status, age and vascular invasion. STG groups patients in the risk groups low, intermediate and high risk (Goldhirsch et al. 2003). Vascular invasion was not available for any patients in the data used in this project. The status was randomly sampled; this may have impaired the performance of STG. The status of vascular invasion may switch the prediction from low to intermediate risk or vice versa.

2.6 Model Evaluation

All models that needed training were trained on a training set and tested on a separate test set. In the pilot study, see section 2.10, 5-fold nested cross validation was used as shown in figure 2.6. The pilot study combined this with forward feature selection[7]. Cross validation and feature selection was not used in the full study because of the computational requirements.

The survival models were evaluated using area under the ROC curve (AUC) and Receiver Operating Characteristics (ROC) described in section 2.6.2 and 2.6.1. Binary classification methods, e.g. receptor inference and 10 year survival prediction, were evaluated using accuracy.

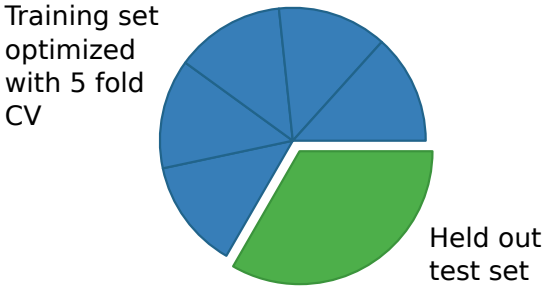

Figure 2.6: Data was partitioned with 33% in the test set and 66% in the training set. In the pilot study the best performance on the training set was found using 5-fold cross validation.

2.6.1 Receiver Operating Characteristic

For a binary classifier the ROC curve is created by calculating the true positive rate and false positive rate at varying discrimination thresholds, see figure 2.7. If one has a high threshold value no positives will be predicted and the true positive rate (tpr) and false positive rate (fpr) will both be 0. As the threshold is lowered all samples will at some point be predicted to be positive and the true positive rate and false positive rate will both be 1.

[7] A short description of forward feature selection by Andrew Ng is available at `http://cs229.stanford.edu/notes/cs229-notes5.pdf`

As shown in figure 2.7 the best classifier will have a ROC curve closer to the top-left corner. A model with random predictions will have a ROC curve that is a straight line from the bottom-left corner to the top-right corner equal to $fpr = tpr$.

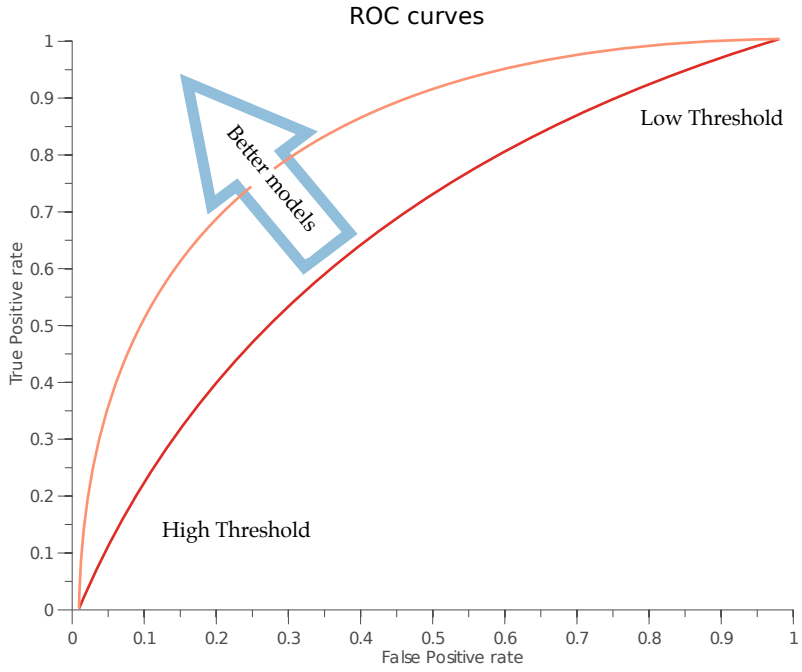

Figure 2.7: Graphical presentation of ROC curves. The line true positive rate = false positive rate is equal to random performance and an AUC of 50%.

2.6.2 Area Under the ROC Curve

The definition of AUC follows Chambless, Cummiskey, and Cui 2011. The AUC is related to the receiver operating characteristic (ROC) curve. The ROC curve plots tpr vs. fpr, the AUC is the area under this curve.

The AUC can be shown to measure the probability of a person having and event is assigned greater risk than a person that did not have an event, formally that is:

$$\text{AUC} = P(Z_i > Z_j | D_i = 0, D_j = 1), \tag{2.38}$$

where Z_i and Z_j are risk scores and D_i and D_j indicate events, 0 means event and 1 means no event.[8]

AUC can be used on survival data with:

$$\text{AUC(t)} = P(Z_i > Z_j | D(t)_i = 0, D(t)_j = 1) \quad (2.39)$$
(Chambless, Cummiskey, and Cui 2011),

here AUC(t) is the AUC evaluated at time t, and $D_i(t)$ and $D_j(t)$ indicate events at time t, 0 means event and 1 means no event at time t. The AUC score ranges from 1 to 0.5. With 1 being perfect prediction and 0.5 being random prediction.

2.7 Inference of Receptor Status

Many of the patients included in the study does not have receptor status measured, this can be seen in table 2.5 p. 35. To solve this it was investigated how receptor status could be inferred from micro array data. Two methods for receptor inference were investigated. The first methods is an unsupervised method based on Gaussian mixtures, explained in section 2.7.1. The second method is an supervised method based on relative gene expression of a varying number of genes, this is explained in section 2.7.2.

2.7.1 Inference of Receptors by Gaussian Mixtures

The Gaussian mixture approach assumes that expression of receptors can be inferred from the expression of a single gene (Lehmann et al. 2011). Karn et al. 2010 used a similar approach as (Lehmann et al. 2011). In a meta-study they collected 3,030 Affymetrix U133A micro arrays. They used the following probes to represent receptor status:

- ER: Affymetrix probe `205225_at` (gene name: Estrogen Receptor 1, entrez: 2099)
- HER2: Affymetrix probe `216836_s_at` (gene name: v-erb-b2 erythroblastic leukemia viral oncogene homolog 2, entrez: 2064)

[8] D is specified different than in (Chambless, Cummiskey, and Cui 2011) to follow the convention used in the MATLAB code.

- PgR: Affymetrix probe `208305_at` (gene name: Progesterone Receptor, entrez: 5241)

For each of the genes a bi-modal Gaussian mixture was fitted using maximum likelihood, the technique is illustrated in figure 2.8. Each sample is assumed to have the receptor expressed if the expression value lies, with highest probability, in the Gaussian component with the highest mean value. Using this method Karn et al. 2010 obtained accuracies 91.6%, 89.2%, and 71.8% for ER, HER2 and PGR respectively. These accuracies where obtained by pooling the data from all the included studies. We investigated

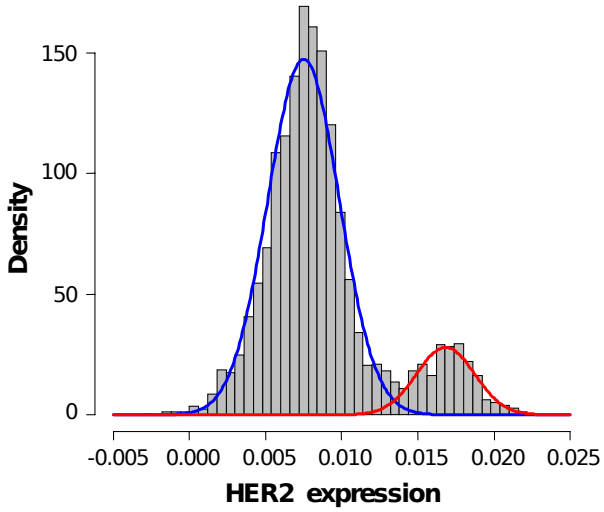

Figure 2.8: Example of Gaussian mixture model. Expression value densities are shown as grey bars. The fitted Gaussian are shown in blue (HER2 negative) and red (HER2 positive). Adapted from (Karn et al. 2010).

if the method used by (Karn et al. 2010) was feasible for multiplatform data. A bi-modal Gaussian mixture was fitted to each of the datasets in Haibe-Kains et al. 2012. We used the `R` package `mclust`[9] for fitting the Gaussian mixtures. Karn et al. 2010 represent the receptors with probes, which is possible because the study only include a single platform, Affymetrix

[9] `http://cran.r-project.org/web/packages/mclust/`

HG U133A. The data from (Haibe-Kains et al. 2012) includes several platforms. Probes were translated to entrez gene ID's. The probes representing the receptors were mapped to entrez ID's (ER:2099, HER2:2064, PgR:5241). To ensure that mixture 1 would have the lowest mean and mixture 2 would have the highest mean a small prior was but on the mean of each cluster. The prior on the mixture means were $0.9 \cdot \text{mean(expression)}$ and $1.1 \cdot \text{mean(expression)}$. The constants where arbitrarily chosen.

The results of fitting bi-modal Gaussians can be seen in section 3.1.1 page 51.

2.7.2 Inference of Receptors by Relative Expression

The following sections describe the top scoring pair (TSP) classifier based on relative gene expression (Geman et al. 2004; Tan et al. 2005). The TSP is based on the expression rank for each gene, not the absolute expression value, as such the method is invariant to most normalization and scaling and comparisons across different platforms are easily performed.

2.7.2.1 Top Scoring Pair Classifier

Marchionni et al. 2013 recently demonstrated the effectiveness of using a TSP for prediction of prognosis in breast cancer patients. The TSP has previously been used to predict ER and BRCA1 status in breast cancer patients (Lin et al. 2009) and for prediction in other types of cancers, See (Eddy et al. 2010). The TSP algorithm works by identifying gene pairs whose expression rank most consistently change between classifications groups. A typical scenario is that one of the genes in a gene pair varies while the other gene is some household gene with relatively constant expression. An example is given in figure 2.9 which show that the expression of entrez ID 2099 vary while the expression of 4953 is relatively constant. The TSP was introduced using a single gene pair. The TSP can be extended to include k gene pairs, here called a k-TSP. For training the k-TSP we follow the procedure used by Marchionni et al. 2013:

1. train a TSP
2. remove the genes in the gene pair from the training data
3. repeat 1 and 2 k times

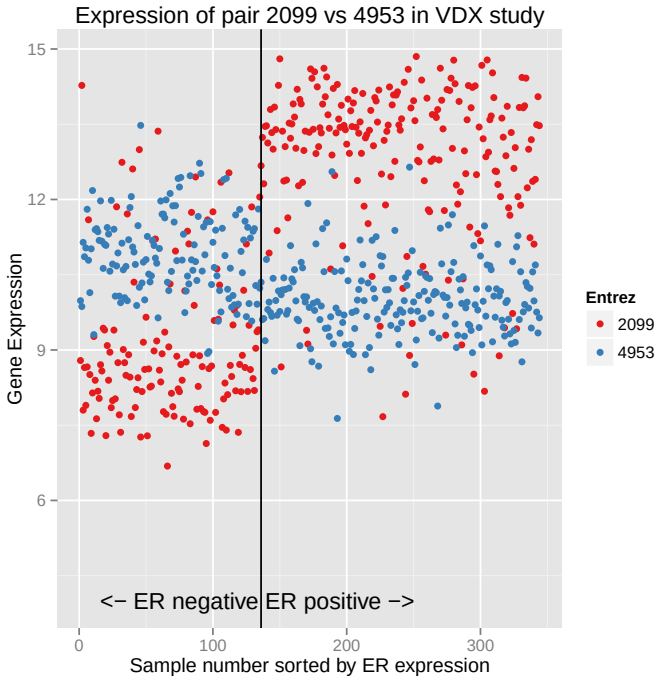

Figure 2.9: Shows patients from the VDX study. Patients are measured to be either ER positives (right of vertical line) or ER negatives (left of vertical line). Expression values of the entrez ids 2099 and 4953 are plotted. The figure illustrates that a gene pair is often composed of varying gene (2099) compared to a gene with more constant expression (4953).

The output from a k-TSP will be a binary matrix of size k by number of samples. Using this matrix the final prediction can be obtained be e.g. majority vote or a threshold value.

2.7.2.2 Training the Top Scoring Pair Classifier

Several implementation of the TSP classifier exists for MATLAB and R (Leek 2009; Magis et al. 2011; Marchionni et al. 2013). We use R code based on the `SwitchBox` R package[10]. The code used for training the k-TSP's in

[10]code available at `http://astor.som.jhmi.edu/~marchion/software`

this project is available in the R package `ktsp`[11], refer to section 2.12 for further details. Training and test set for the receptors ER, HER2 and PgR was extracted from the data available in (Haibe-Kains et al. 2012). Exact settings for extraction of datasets can be seen in the R package `datathesis`[12] help files for `er-random`, `her2-random` and `pgr-random`. For each receptor we extracted samples with the receptor status measured with either immunohistochemistry (IHC) or FISH. For all receptors 50% of the data was randomly assigned to the test set. For each receptor we identified the 50 TSP's. Using 5-fold cross validation the optimal number of pairs, k, was chosen. At each number of pairs all possible threshold values were tested using cross validation. Using the identified k a k-TSP was trained using the entire training set. Finally the prediction accuracy was evaluated at the test set. Missing expression values where imputed using KNN impute from the R `impute` package (*impute: impute: Imputation for microarray data*). The number of clusters was set to 10 and other settings where left at default values. For all receptor inference, the following platforms were included: agilent, affy, affy.u95, agilent99. Table 2.2 shows the available data for training and testing the k-TSP's. Section 3.1.2 presents the results of receptor inference by k-TSP.

Table 2.2: Patients available for training and testing the k-TSP for prediction of status of different receptors.

	Available	Negatives	Positives
ER	4084	1127	2957
HER2	1350	980	370
PgR	2100	878	1222

2.8 Microarray Derived Features

Several micro array derived features have been used for predicting recurrence risk in breast cancer patients, among others:

1. PAM50 (Parker et al. 2009)

[11] `ktsp` code available at: `https://bitbucket.org/skaae/ktsp`
[12] `datathesis` code available at: `https://bitbucket.org/skaae/datathesis`

2. OncotypeDX (Paik et al. 2004)
3. MammaPrint (Veer et al. 2002)

PAM50 classifies breast cancer tumors into the subtypes: HER2 enriched, basal-like, luminal B and luminal A. OncotypeDX predicts the risk of distant recurrence after 10 years in node negative, estrogen receptor positive patients. The prediction is based on the expression in 21 genes (Paik et al. 2004). MammaPrint predicts 5-year metastatic recurrence as good or bad by using the expression of 70 genes (Veer et al. 2002; Vijver et al. 2002). Some of these molecular features have been developed for patient sub groups, and they might perform poorly in the patient population used in this study.

All of the above mentioned features rely on at least some micro array probes, which do not map to a Entrez ID. Because this project uses Entrez Id's to map between different platforms it is difficult to use these methods. (Marchionni et al. 2013) has recently shown that the MammaPrint assay can be accurately reproduced by use of k-TSP. A k-TSP was trained for predicting recurrence free survival 10 years. The available data is identical to the data shown in table 2.5. Exact settings for data extraction can be seen in the `datathesis` package help file for `surv10`. Table 2.3 shows the number of patients with and with out recurrence. Because the number of patients with out recurrence is larger than the number of patients with recurrence we rescale the classes in the training data to equal size.

Table 2.3: Patients available for training and testing the k-TSP for prediction of recurrence at 10 years

	Available patients	Without Recurrence	Recurrence
10 year survival	1374	864	510

2.9 Data Collection

This section describes the data used in the project. Section 2.9.1 describes the data inclusion criteria. Section 2.9.2 describes the used covariates, normalization and handling of missing expression values. The included data is

presented in section 2.9.3 and visualized using Kaplan-Meier plots in section 2.9.4.

2.9.1 Data Inclusion Criteria

Breast cancer data from Haibe-Kains et al. 2012 was used in the project. The following inclusion criteria where used:

1. Either recurrence free survival or distant metastasis free survival time must be available. If both are available recurrence free survival is preferred.
2. Tumor size, histological grade, nodal involvement, patient age and treatment status must be available.
3. Microarray data must be publicly available and measured with either Agilent, Affymetrix or Illumina platforms.
4. If receptor status was measured with either IHC or FISH this value was preferred otherwise the inferred receptor status value was used.

2.9.2 Covariates, Normalization and Missing Values

The following covariates were included in the model:

1. age [Continuous, years]
2. her2 receptor status, [Categorical, -1/1]
3. Estrogen receptor status [Categorical, -1/1]
4. PgR receptor status [Categorical, -1/1]
5. Nodal involvement [binary, -1/1]
6. Tumor size [Continuous, cm]
7. Histological grade [Ordinal, 1/2/3]
8. Treatment [Categorical, -1/1]
9. Predicted recurrence at 5 years [Categorical, -1/1]
10. Predicted recurrence at 10 years [Categorical, -1/1]

Continuous and ordinal covariates were scaled to zero mean and unit variance. Categorical covariates where represented as ±1 if the number of categories was equal to 2 otherwise one hot encoding.

For genomic data missing values where calculated with KNN impute from the `impute` R package (Hastie et al. 1999). The imputation of missing values was performed using 10 clusters and all other settings as default. Data was collected from different micro array platforms. A shared gene set was created by mapping probes to entrez gene ID's. Several probes may map to the same Entrez ID, in these cases the probe with the highest variance within each platform was chosen. Only entrez ID's shared across all used platforms were considered.

2.9.3 Included Data

This section presents the data used for training of survival models. Table 2.4 shows an overview of collected data. Table 2.5 shows basic statistics of the included data. The total number of included patients is 2064. The patients included in table 2.5 were selected based on the criteria listed in section 2.9. Note that many of the patients does not have HER2 and PgR status measured, which made it necessary to infer these from micro-array data.

Table 2.4: Microarray studies. Collected from (Haibe-Kains et al. 2012). Os: overall survival, Rfs: Recurrence free survival, dmfs: Distant metastasis free survival. Refer to Haibe-Kains et al. 2012 for references on the specific datasets.

Study	Patients	Os	Rfs	Dmfs	Platform
NKI	337	yes	no	yes	agilent
STNO2	118	yes	yes	no	cdna.stanford
NCI	99	no	yes	no	cdna.nci
KOO	88	no	no	no	affy.u95
MSK	99	no	no	yes	affy
UPP	251	no	yes	no	affy
STK	159	no	yes	no	affy
VDX	344	no	no	yes	affy
UNT	133	no	no	yes	affy
MAINZ	200	no	no	yes	affy
DUKE	171	yes	no	no	affy.u95
DUKE2	160	no	no	no	affy.3x
CAL	118	yes	no	yes	affy
TRANSBIG	198	yes	no	yes	affy
EMC2	204	no	no	yes	affy
LUND	143	no	no	no	swegene
LUND2	105	no	no	no	swegene
FNCLCC	150	no	no	no	umgc.ircna
MDA4	129	no	no	no	affy
NCCS	183	no	no	no	affy
IRB	129	no	no	no	affy
DFHCC	115	no	no	yes	affy
DFHCC2	84	no	no	no	affy
EORTC10994	49	no	no	no	affy
HLP	53	no	no	no	illumina
MAQC2	230	no	no	no	affy
MCCC	75	no	no	no	illumina
MUG	152	no	no	no	operon
DFHCC3	40	no	no	no	affy
PNC	92	yes	yes	no	affy
EXPO	353	no	no	no	affy
UCSF	162	yes	no	yes	cdna.ucsf
UNC4	305	yes	yes	no	agilent99
SUPERTAM_HGU133A	856	no	no	yes	affy
SUPERTAM_HGU133PLUS2	517	no	no	yes	affy

Table 2.5: Basic statistic of data used for training and evaluation of survival models. n=2064.

Age (years)	
Median	54 (24-91)
Mean	56 (±13.16)
No age inf.	0
Size (cm)	**Patients**
<=1.5	583
(1.5-2.5]	866
>2.5	615
No size inf.	0
Nodal Involvement	
0	1451
1	613
No Node inf.	0
Histological Grade	
1	389
2	895
3	780
No Grade inf.	0
Estrogen Receptor Status	
Negative	439
Positive	1617
No ER inf.	8
HER2 Status	
Negative	213
Positive	65
No HER2 inf.	1786
Progesterone Receptor Status	
Negative	257
Positive	579
No PgR inf.	1228
Treatment Status	
Treatment	1037
No Treatment	1027
No Treatment data inf.	0

2.9.4 Kaplan-Meier Plots

Kaplan-Meier plots were used to plot survival curves for patients with different characteristics. Briefly a Kaplan-Meier plot shows the probability of having recurrence at different times. A group with high risk of recurrence will have a curve that decreases fast and visa versa for group with low risk of recurrence. Significant difference between groups where tested using log-rank test (Zeileis et al. 2008) with a p-value below 0.05 considered significant. Figure 2.10, 2.11 and 2.12 show Kaplan-Meier plot for patients with different clinical parameter used for stratification, the shaded areas indicate the two sided 0.95% confidence intervals. The patients used in the figures are identical to the patients used for table 2.5. Figure 2.10 shows that increased histological grade, NPI, STG and tumor size are associated with decreased time recurrence, which is also reflected in low p-values. Figure 2.11 shows increased risk for ER and PgR negative patients, both having significant p-values. HER2 positive receptor status seems to be associated with slightly higher risk, even though the p-value is not significant. Treatment does seem to be predictive of recurrence risk, the p-value is not significant. Lastly figure 2.11 shows that nodal involvement is associated with higher recurrence risk, the p-value is significant. The prediction of survival at 10 years does not predict survival and the p-value is not significant. Low age seems to be associated with higher risk of recurrence as seen in figure2.11, the p-value is significant.

Table 2.6 shows classification by NPI and St. Gallen, used for creating figure 2.10. The table shows that both methods classify the majority of patients in the intermediate risk group, indicating that the methods are not that useful for stratification of patients.

Table 2.6: Risk stratificatoin by NPI and St. Gallen 2006

	Good	Intermediate	Poor	Not Available
NPI	634	1251	179	0
STG	72	1878	114	0

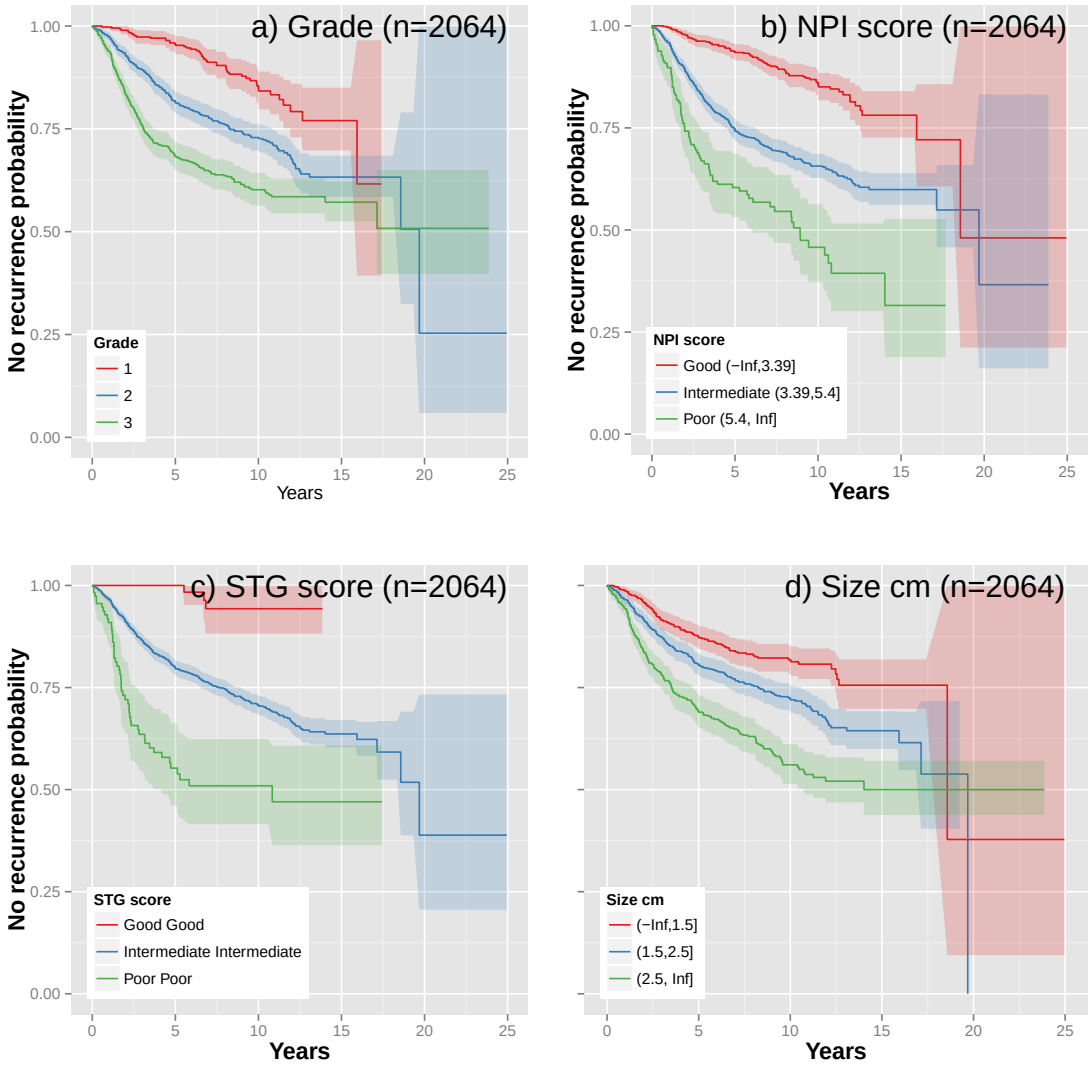

Figure 2.10: Kaplan-Meier plot of included data. Panel a) stratification by histological grade (p-value: $<$ 2.22e-16), b) stratification by NPI (p-value: $<$ 2.22e-16), c) stratification by St. Gallen 2006 (p-value: 1.7263e-10), d) stratification by tumor size (p-value: 9.992e-16).

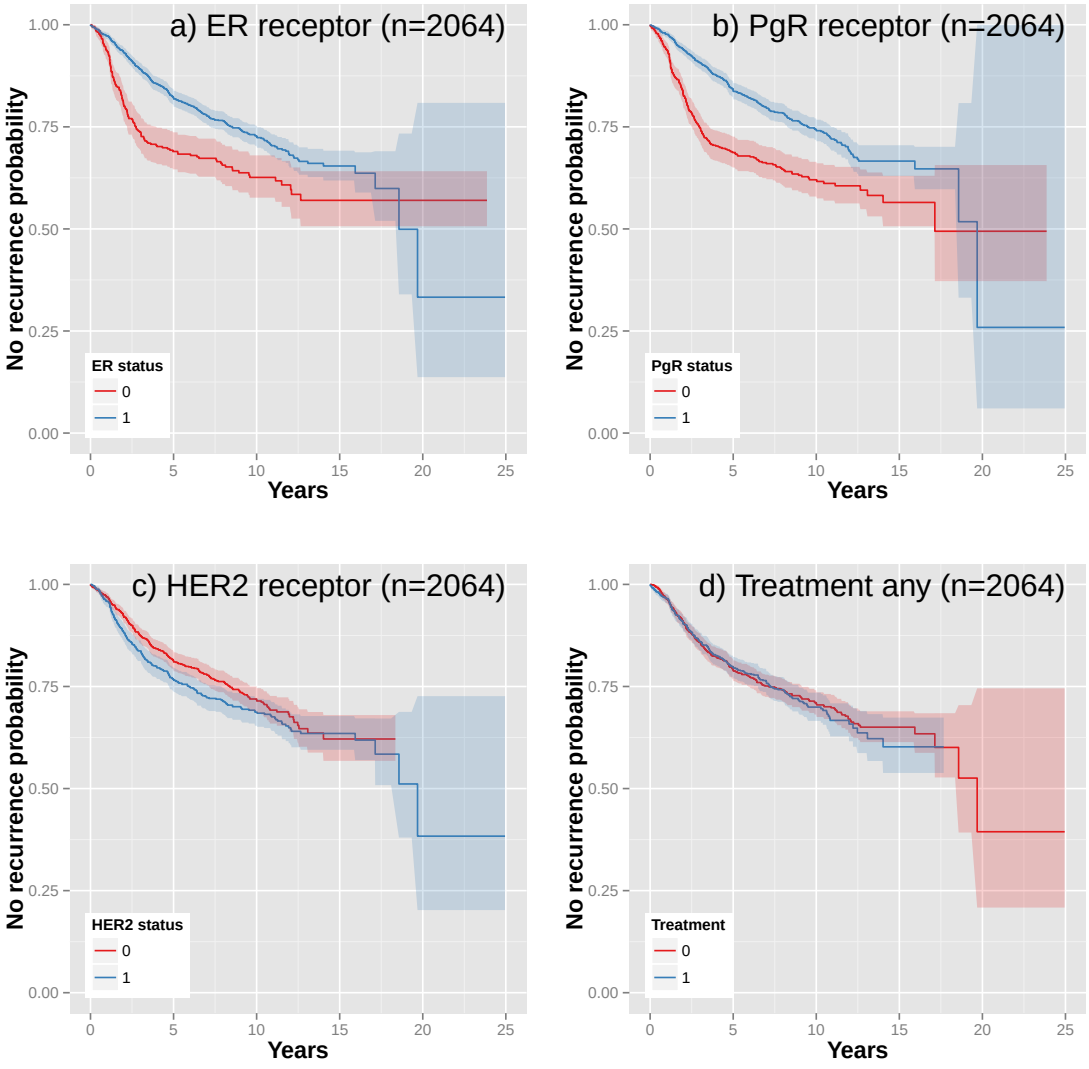

Figure 2.11: Kaplan-Meier plot of included data. Panel a) stratification by ER (p-value: 1.7784e-06), b) stratification by PgR (p-value: 3.4957e-10), c) stratification by HER2 (p-value: 0.068235), d) stratification by Treatment status (p-value: 0.72859). For receptors the inferred receptor status was used if IHC or FISH measurements was unavailable.

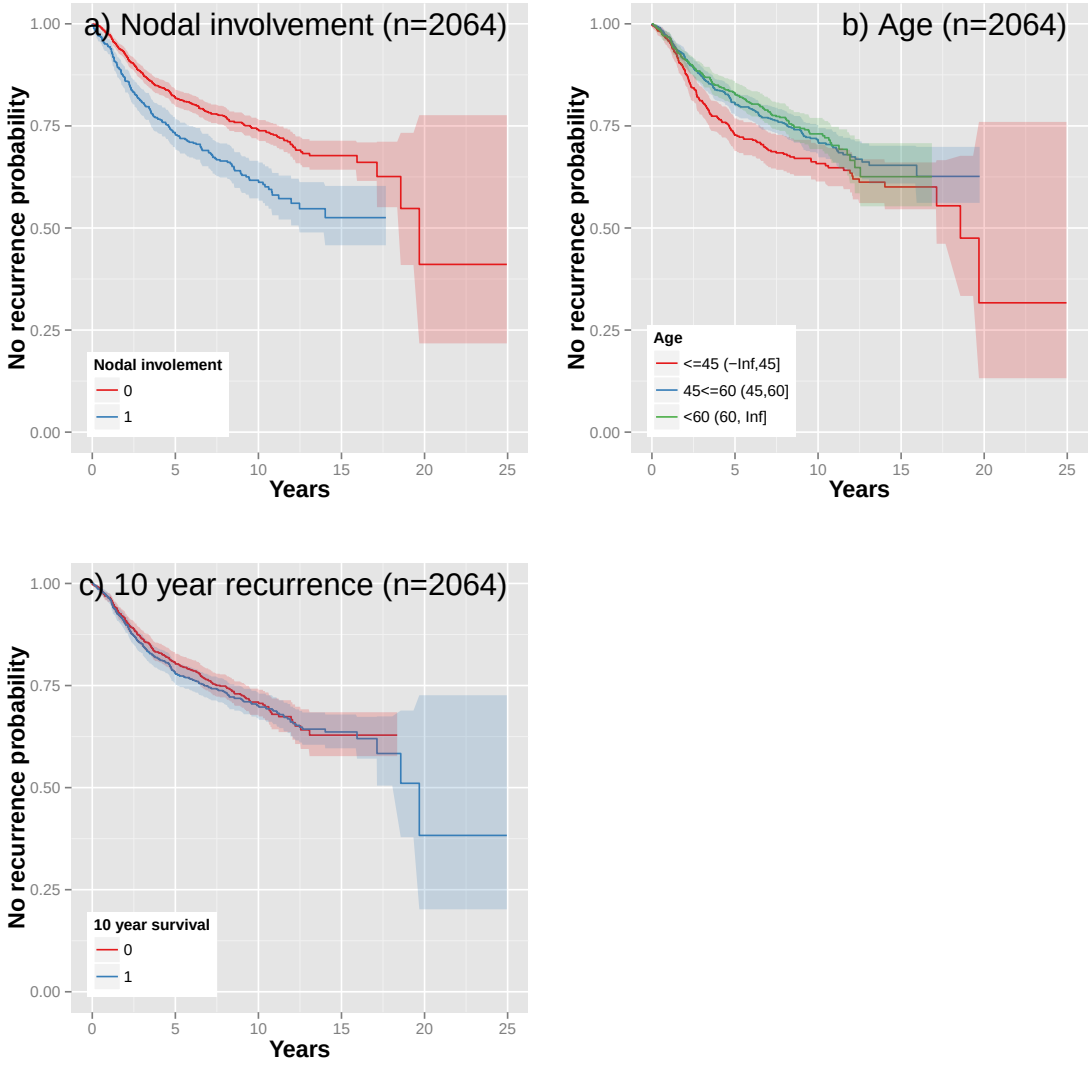

Figure 2.12: Kaplan-Meier plot of included data. Panel a) stratification by nodal involvement (p-value: 1.2167e-07), b) stratification by age (p-value: 0.011539), c) stratification by 10 year survival prediction. 0 is event and 1 is right censoring at 10 years (p-value: 0.59559).

2.10 Pilot study

The pilot study was used to determine if the GP based survival models and RSF models performed better than existing survival models. The pilot compared the performance of GP models, RSF models, CoxPH models, NPI and STG. The pilot study includes covariates that were readily available at the beginning of the project, that includes samples from both the Haibe-Kains et al. 2012 and Curtis et al. 2012 (METABRIC). METABRIC was because at the time the pilot study was performed it was thought that the genomic data from METABRIC would be released. The pilot study also differs in outcome variable. The pilot study uses overall survival, but the final study uses recurrence free survival or distant metastasis free survival. The outcome variable was changed because of the limited amount of "overall survival samples" when METABRIC was not included. Lastly the pilot study used forward feature selection, which is not used in the final evaluation.

The following covariates where used: age, nodal involvement, tumor size, histological grade and treatment information. Table 2.7 shows the size of the dataset used in the pilot study. All samples with NA values were removed prior to training and testing.

Table 2.7: Samples in training and test set used in pilot study. 33% of the samples were assigned to the test set.

Training	Test	total
1400	690	2090

2.10.1 Models

GP models were trained using the `gpstuff` MATLAB package (Vanhatalo et al. 2013). A neural-network covariance function combined with constant covariance function was used. Weights variance was either free in all dimensions (*free*) or shared between dimensions (*not free*). For each model the

features where selected using forward selection and 5 fold nested cross validation. The neural network plus constant covariance function was chosen after several alternatives had been explored, among others Matern, squared exponential and polynomial covariance and various combinations of these, results not shown. The neural network plus constant covariance function generally performed best which is the same result as obtained by Joensuu et al. 2012.

The R package `randomForestSRC` was used for training RSF models (Ishwaran and Kogalur 2013). A forest of 1000 trees was grown using the training data. Minimum terminal node size, number of candidate features at splits and number of considered split values were considered hyper parameters to the model. CoxPH models were evaluated using `coxphfit` from the MATLAB statistics toolbox. NPI and STG models were ported to MATLAB from the implementation available in the `genefu` R package (Haibe-Kains et al. 2013). Forward feature selection was used for GP models and CoxPH models using `sequentialfs` from the MATLAB statistics toolbox.

2.10.2 Results

AUC scores for the different models are reported in table 2.7. Table 2.7 shows that the GP models generally performed best (AUC Feat selection column). The *free* and *not free* GP models practically performed on par. The *not free* GP model was evaluated on the test set. The AUC evaluated at 10 years was 73.03% on the test set. The test AUC is shown in figure 2.14. The test AUC scores for NPI and St. Gallen 2006 are 69.24% and 56.40% respectively.

Figure 2.13 shows conditional plots of the GP model. The conditional plots were created by tying all parameters but the selected at their mean value. The conditional plot shows how varying a single parameter influences the prediction. Figure 2.13 generally shows the expected, e.g. increased tumor size and histological grade are associated with increased risk. Interestingly young age seems to be associated with increased risk.

Table 2.8: Final AUC scores evaluated at 10 years. AUC feature selection is the score found using 5-fold cross validation of the training set. AUC train is the AUC score from fitting a model using the features found by forward selection. Columns 3 to 12 indicate if the feature was selected by forward feature selection. The features Treat. RT/CT/HT are only available in Curtis et al. 2012 and are not included in the final study. RT: radio therapy, CT: chemo therapy, HT: hormonal therapy.

Model name	AUC Feat selection CV test	AUC Train	Age	Nodes	Tumor size	Grade	Her2	ER	Treat. any	Treat. RT	Treat CT	Treat. HT
GP free	0.6971	0.7202	1	1	1	1	1	0	1	0	1	0
GP Not Free	0.6994	0.7217	1	1	1	1	1	0	1	0	0	0
CoxPH	0.6759	0.6646	1	1	1	1	0	0	1	0	0	1
RFS	0.6552	0.6552	NA	→								

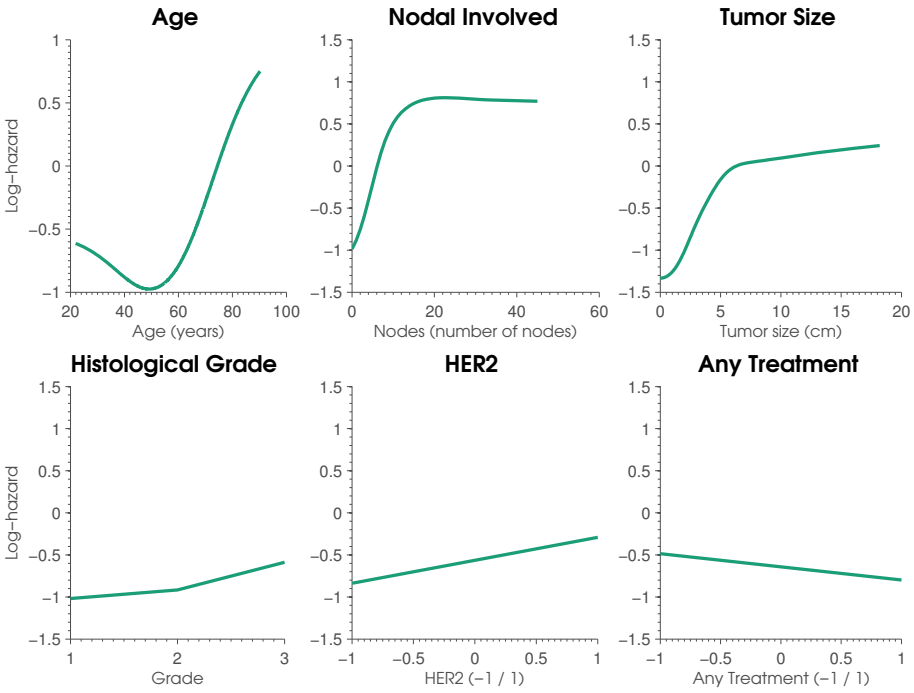

Figure 2.13: Conditional plots of covariates in GP model, features selected by forward feature selection. (Evaluated at training data)

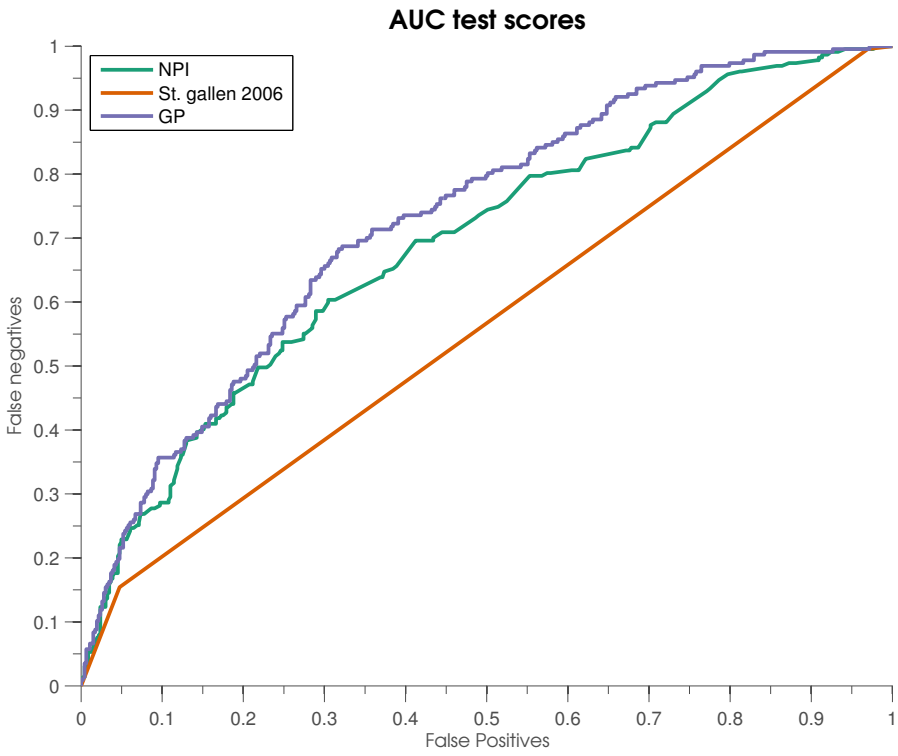

Figure 2.14: Test AUC scores in pilot study.

2.11 Evaluated Models in Final Study

The pilot study showed that the GP based models generally performed best. Further investigations are limited to GP based models in comparison with CoxPH, NPI and STG models. Feature selection was not performed because of the computational times involved. The outcome variable in the final study was changed from overall survival to recurrence free survival (rfs) or distant metastasis free survival (dmfs), because few samples with overall survival status is available in Haibe-Kains et al. 2012. The selected models were evaluated at the datasets:

1. Baseline model: age, histological grade and tumor size and treatment status

2. Receptor model: baseline + HER2, ER, PgR
3. Fingerprint model: receptor + fingerprints

Fingerprints means features derived from microarray data. The models will be trained using the data in table 2.5 p. 35. 33% of the samples will be put in a held out test set. For all models identical training and test sets are used. The performance criterion is AUC evaluated at 10 years. The result for the different models are presented in section 3.3.

2.12 Code

The code used in this project is programmed in R, MATLAB and Sweave, described in section 2.12.1, 2.12.2 and 2.12.3 respectively. Except for the pilot study all analysis, figures and tables should be reproducible using the code accompanying this project. Figures and tables are either generated in R or MATLAB. Table 2.9 shows what code was used to generate which figures and tables.

2.12.1 R code

Most of the R and Sweave code depends on the R packages `ktsp` and `datathesis` that were written during the project. The source for both packages are available on `https://bitbucket.org/skaae/`. To install the R packages you need a to install R, see `http://cran.r-project.org/` for installation instructions for most operating systems. When R is installed the `ktsp` and `datathesis` can be installed from the online repository using the code shown in listing 2.1.

Listing 2.1: Code for installing R packages

```
install.packages("devtools")   #package for installing from ↩
    bitbucket
require("devtools")
install_bitbucket("datathesis","skaae")  #install datathesis ↩
    from bitbucket
install_bitbucket("ktsp","skaae")  #install ktsp from bitbucket
require(datathesis)
require(ktsp)
```

The `datathesis` package contains all data sets used in the rapport. The following datasets are included in the package:

- `matlabsurv` data used for training survival models.
- `er-random`: Data set used for training ktsp for prediction of Estrogen receptor.
- `her2-random`: Data set used for training ktsp for prediction of HER2 receptor.
- `pgr-random`: Data set used for training ktsp for prediction of PgR receptor.
- `allclinical`: All clinical data

The help file for each dataset contains code for reproducing the dataset. Access the help file by running `help("dataset-name")` from the R console. To reproduce the data sets you need to download the micro array data which is available at `https://bitbucket.org/skaae/thesis_sweave` as a zip-file, see listing 2.2. In each of the examples you need to change the variable `path.haibekains` to the folder where the folder `breast_cancer/datasets/haibekains` is located. The folder is a part of the `thesis_sweave` repository.

2.12.2 MATLAB code

MATLAB and Sweave code is available as bitbucket repository at: `https://bitbucket.org/skaae/thesis_sweave`. To download the code locally run the code in listing 2.2. Download `wgethttp://compbio.dfci.harvard.edu/pubs/sbtpaper/data.zip` and unzip file to `datasets/haibekains/` in the cloned repository. The `*.RData` files from the zip file must be in `datasets/haibekains/` not in any subfolder.

Listing 2.2: Code for cloning Sweave and MATLAB code

```
git clone https://skaae@bitbucket.org/skaae/thesis_sweave.git
```

The Matlab code for training survival models is available in the folder: `matlab_files/serverfiles/`. MATLAB code uses data from the `matlabsurv` data set in the `datathesis` package. If you need to recreate the csv files run the example code in the help to the `matlabsurv` data set. The csv

files for the training and test sets are saved to the folders specified in `file.matlab.train` and `file.matlab.test` respectively. To train the survival models run the script `matlab_files/serverfiles/evaluate_all_models.m`. The csv files with training and test data need to be in the same folder as `evaluate_all_models.m`. The MATLAB code uses GPstuff 4.3 for running GP survival models[13]. All code was run using MATLAB version 2013b.

2.12.3 Sweave Code

The rapport is generated using Sweave(Leisch 2002), i.e LaTeXwith embedded R code. To generate the rapport download RStudio[14] and LaTeX[15]. Download the `datathesis` package and the `ktsp` package as described in section 2.12.1. Download the Sweave code from bitbucket using the code given in listing 2.2. This will download the Sweave code to your current directory. Unzip the file. Within this directory open the `master.Rnw` file in RStudio and push the compile PDF. This will generate the rapport as `master.pdf`.If you are not on a MAC obtain sweave from `http://www.stat.uni-muenchen.de/~leisch/Sweave/` and follow the instructions.

2.12.4 R Session Information

- R version 3.0.2 (2013-09-25), `x86_64-apple-darwin10.8.0`
- Locale: `en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, graphics, grDevices, grid, methods, parallel, splines, stats, utils
- Other packages: affy 1.39.4, AnnotationDbi 1.23.27, Biobase 2.21.7, BiocGenerics 0.7.5, BiocInstaller 1.12.0, biomaRt 2.17.3, cacheSweave 0.6-1, caret 5.17-7, caTools 1.14, class 7.3-9, cluster 1.14.4, coin 1.0-23, data.table 1.8.10, datathesis 0.1.0, DBI 0.2-7, dplyr 0.1.2, e1071 1.6-1, filehash 2.2-1, foreach 1.4.1, gdata 2.13.2, genefu 1.11.0,

[13] `http://becs.aalto.fi/en/research/bayes/gpstuff/`
[14] https://www.rstudio.com/
[15] http://latex-project.org/ftp.html

ggplot2 0.9.3.1, gplots 2.11.3, gridExtra 0.9.1, gtools 3.1.0, impute 1.35.0, KernSmooth 2.23-10, ktsp 0.1.0, lattice 0.20-23, MASS 7.3-29, mclust 4.2, minpack.lm 1.1-8, nnet 7.3-7, org.Hs.eg.db 2.10.1, plyr 1.8, prodlim 1.3.7, qpcR 1.3-7.1, R.matlab 2.1.0, randomForest 4.6-7, RColorBrewer 1.0-5, reshape2 1.2.2, rgl 0.93.996, robustbase 0.9-10, ROCR 1.0-5, RSQLite 0.11.4, stashR 0.3-5, stringr 0.6.2, survcomp 1.11.0, survival 2.37-4, xtable 1.7-1

- Loaded via a namespace (and not attached): affyio 1.29.5, amap 0.8-7, assertthat 0.1, bitops 1.0-6, bootstrap 2012.04-1, codetools 0.2-8, colorspace 1.2-4, dichromat 2.0-0, digest 0.6.3, gtable 0.1.2, IRanges 1.19.38, iterators 1.0.6, labeling 0.2, modeltools 0.2-21, munsell 0.4.2, mvtnorm 0.9-9997, preprocessCore 1.24.0, proto 0.3-10, R.methodsS3 1.5.2, R.oo 1.15.8, R.utils 1.28.4, Rcpp 0.11.0, RCurl 1.95-4.1, rmeta 2.16, scales 0.2.3, stats4 3.0.2, SuppDists 1.1-9, survivalROC 1.0.3, tools 3.0.2, XML 3.95-0.2, zlibbioc 1.7.0

2.12.5 Scripts

Table 2.9 shows what code that was used to generate which figures and tables.

Table 2.9: List of scripts used to create figures and tables

Figure / Table	Script
Figure 2.1 p. 11	`matlab_files/plot_fevent_surv_haz.m`
Table 2.2 p. 30	`documents/methods_infer_receptors.Rnw`
Table 2.3 p. 31	`documents/methods_fingerprints.Rnw`
Table 2.4 p. 34	`table/table_combined_data.R`
Table 2.5 p. 35	`documents/methods_data_overview.Rnw`
Table 2.6 p. 36	`documents/methods_kaplan_meier.Rnw`
Figure 2.10 p. 37	`documents/results_kaplan_meier.Rnw`
Figure 2.11 p. 38	`documents/results_kaplan_meier.Rnw`
Figure 2.12 p. 39	`documents/results_kaplan_meier.Rnw`
Figure 3.1 p. 53	`R_files/bimodal_plots.R`
Figure 3.2 p. 54	`R_files/bimodal_plots.R`
Figure 3.3 p. 55	`R_files/bimodal_plots.R`
Table 3.2 p. 56	`documents/results_infer_receptors_TSP.Rnw`
Figure 3.4 p. 57	`R_files/ktsp_receptors.R`
Table 3.3 p. 58	`R_files/ktsp_receptors.R`
Figure 3.5 p. 60	`R_files/ktsp_receptors.R`
Table 3.4 p. 59	`R_files/ktsp_receptors.R`
Figure 3.6 p. 61	`R_files/ktsp_receptors.R`
Table 3.5 p. 61	`R_files/ktsp_receptors.R`
Table 3.6 p. 62	`documents/results_surv10.Rnw`
Figure 3.7 p. 63	`R_files/ktsp_receptors.R`
Table 3.7 p. 62	`R_files/ktsp_receptors.R`
Table 3.8 p. 64	`matlab_files/evaluate_all_models.m`
Figure 3.8 p. 65	`matlab_files/evaluate_all_models.m`
Figure 3.9 p. 66	`matlab_files/evaluate_all_models.m`
Figure 3.10 p. 67	`matlab_files/evaluate_all_models.m`
Figure 3.11 p. 68	`matlab_files/evaluate_all_models.m`
Table 5.4 p. 82	`R_files/ktsp_receptors_tables.R`
Table 5.5 p. 83	`R_files/ktsp_receptors_tables.R`
Table 5.6 p. 84	`R_files/ktsp_receptors_tables.R`
Table 5.7 p. 85	`R_files/ktsp_receptors_tables.R`

Chapter 3

Results

This chapter presents the results of the project. The chapter mainly tables and figures presenting the results. Discussions of the results are in chapter 4. This chapter starts with the results of receptor inference presented in section 3.1. The results of predicting 10 year recurrence from micro array data is shown in section 3.2. Finally section 3.3 concludes the chapter with the results of modelling recurrence risk in breast cancer patients.

3.1 Infer Receptors

Results of inference of receptors by Gaussian mixture models are described in section 3.1.1, results of inference by k-TSP's are described in section 3.1.2.

3.1.1 Receptor Inference Using Gaussian Mixture Models

Figure 3.1, figure 3.2 and figure 3.3 show the attempt to fit a bimodal Gaussian mixture to each of the datasets. The performance for each receptor is given in table 3.1. Using the Gaussian mixture models all methods perform poorly, with only inference of ER being better than random guessing. Figure 3.1 shows that the expression value density for entrez ID 2099 (ER) has a bimodal distribution for all datasets except DUKE. Note that booth SUPERTAM studies have close to zero ER negative patients and therefor should not be bimodal. Figure 3.2 shows bimodal Gaussian mixtures fitted to expression densities of entrez ID 2064 (HER2). The MSK study has

Receptor	Test Performance
ER	0.591
HER2	0.457
PgR	0.493

Table 3.1: Accuracy of Gaussian mixture models for inference of receptors.

no HER2 negatives. Several studies fail to show bimodality, e.g. MAQC2. Figure 3.3 shows bimodal Gaussian mixtures fitted to expression values of entrez ID 5241 (PgR). SUPERTAM HGU133A and UNT only have a small fraction of PgR positives. Several studies fail to bimodality, e.g. UNC4, KOO and DUKE.

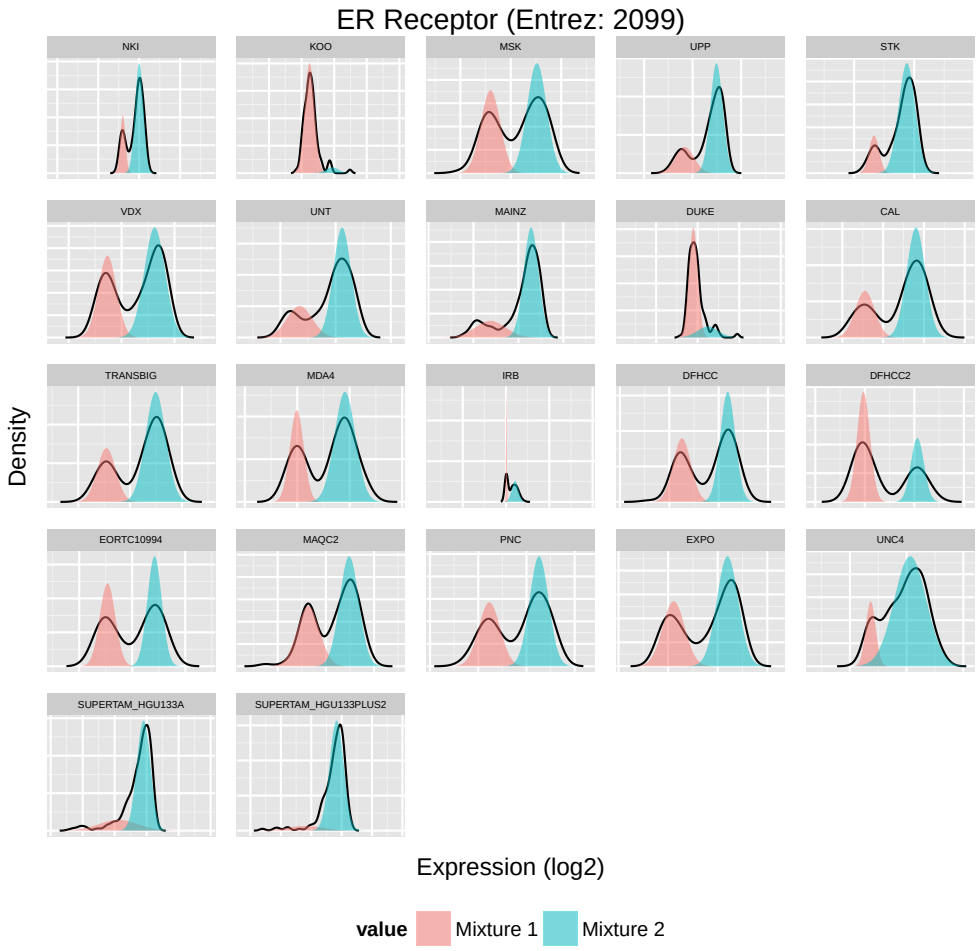

Figure 3.1: Density plot of Entrez ID 2099 with fitted Gaussian overlaid.

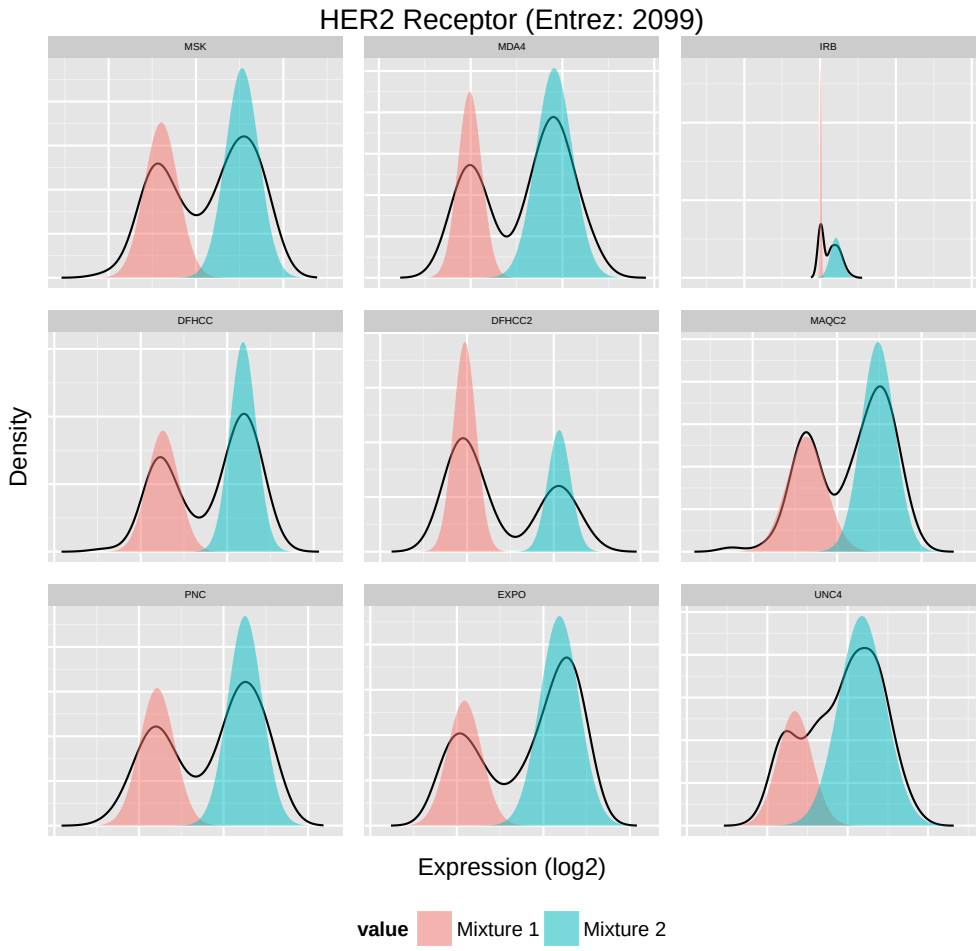

Figure 3.2: Density plot of Entrez ID 2064 with fitted Gaussian overlaid.

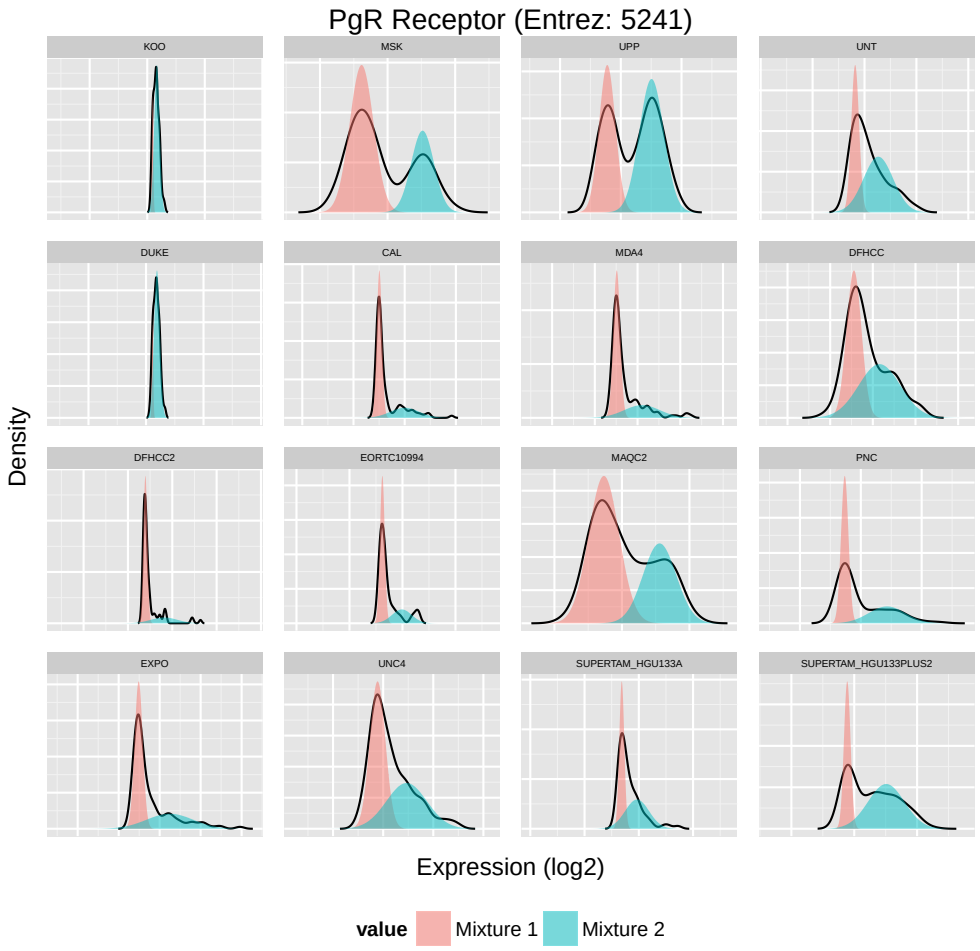

Figure 3.3: Density plot of Entrez ID 5241 with fitted Gaussian overlaid.

3.1.2 Receptor Inference Using Top Scoring Pair

The following section shows the results of receptor inference using k-TSP's. The final results are listed in table 3.2. CV test performance and training performance are shown in figure 3.4, 3.5 and 3.6 for ER, HER2 and PgR respectively. Included gene pairs are listed in table 3.3, 3.4 and 3.5 for ER, HER2 and PgR respectively.

Table 3.2: Final accuracy of receptor inference using k-TSP. Threshold is the voting threshold when combining the TSP's.

	Best k	Threshold	Training Accuracy	Test Accuracy
ER	30	17	0.9094	0.8939
PgR	8	3	0.8385	0.7960
HER2	9	4	0.7910	0.7908

3.1.2.1 ER

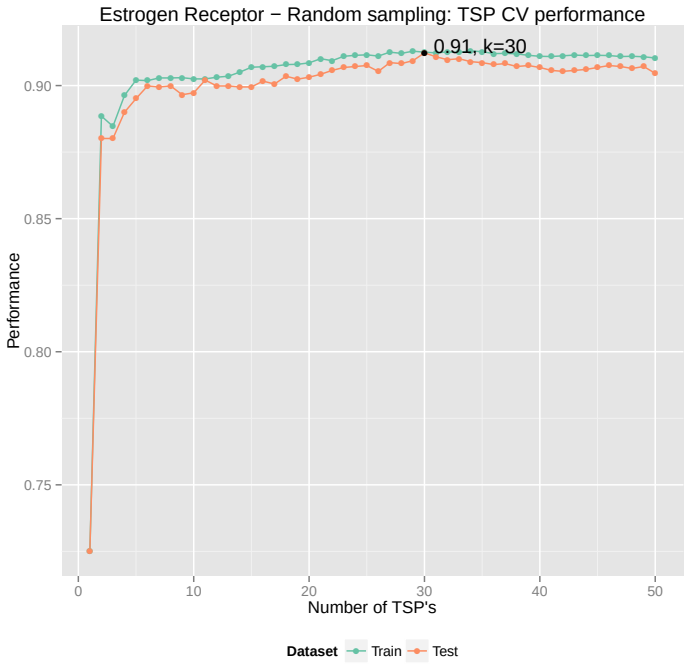

Figure 3.4: Accuracy scores for k-TSP's as k is varied from 1 to 50. Train and test accuracy is the mean CV accuracy.

Entrez A	Hugo A	Entrez B	Hugo B
2099	ESR1	4953	ODC1
2625	GATA3	8884	SLC5A6
23158	TBC1D9	114908	TMEM123
771	CA12	5214	PFKP
7802	DNALI1	6491	STIL
9	NAT1	9111	NMI
18	ABAT	10926	DBF4
4602	MYB	2195	FAT1
7033	TFF3	6590	SLPI
7031	TFF1	991	CDC20
8416	ANXA9	9212	AURKB
2066	ERBB4	53335	BCL11A
4137	MAPT	898	CCNE1
10551	AGR2	6280	S100A9
9687	GREB1	6664	SOX11
8614	STC2	1001	CDH3
2203	FBP1	8140	SLC7A5
7494	XBP1	7388	UQCRH
51097	SCCPDH	8833	GMPS
1602	DACH1	51442	VGLL1
8382	NME5	1058	CENPA
10974	ADIRF	445	ASS1
6337	SCNN1A	5918	RARRES1
22977	AKR7A3	11004	KIF2C
3480	IGF1R	2296	FOXC1
2674	GFRA1	9420	CYP7B1
2879	GPX4	4904	YBX1
3249	HPN	1054	CEBPG
79921	TCEAL4	7913	DEK
6478	SIAH2	9833	MELK

Table 3.3: Genes in Best k-tsp for ER named by Entrez ID and Hugo Id

3.1.2.2 HER2

Entrez A	Hugo A	Entrez B	Hugo B
10948	STARD3	596	BCL2
2886	GRB7	1108	CHD4
51755	CDK12	23131	GPATCH8
2064	ERBB2	6659	SOX4
5709	PSMD3	6651	SON
8564	KMO	9639	ARHGEF10
5409	PNMT	2145	EZH1
8714	ABCC3	23189	KANK1

Table 3.4: Genes in Best k-tsp for HER2 named by Entrez ID and Hugo Id

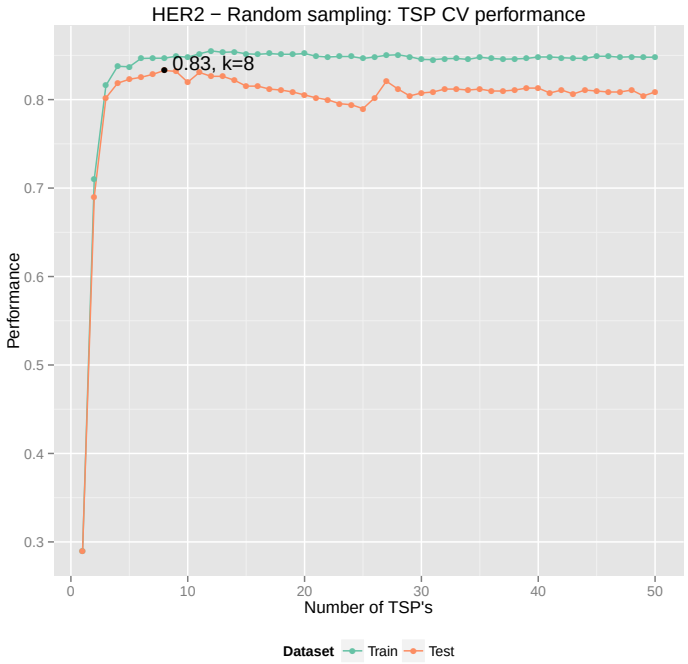

Figure 3.5: AUC scores for k-TSP's as k is varied from 1 to 50. Train and test AUC is the mean CV accuracy.

3.1.2.3 PgR

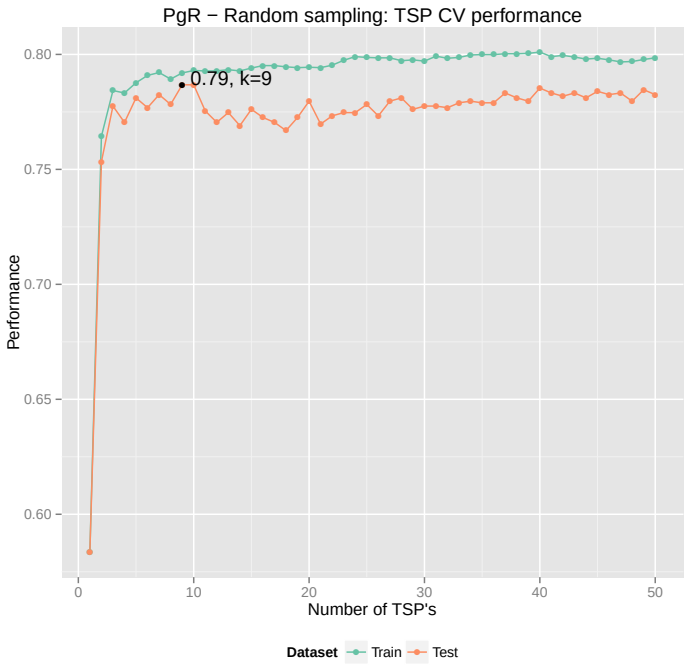

Figure 3.6: Accuracy scores for k-TSP's as k is varied from 1 to 50. Train and test accuracy is the mean cross validation accuracy.

Entrez A	Hugo A	Entrez B	Hugo B
771	CA12	4860	PNP
2099	ESR1	4704	NDUFA9
2625	GATA3	11130	ZWINT
7802	DNALI1	11339	OIP5
9687	GREB1	7272	TTK
9	NAT1	1475	CSTA
18	ABAT	9833	MELK
2066	ERBB4	1054	CEBPG
4137	MAPT	898	CCNE1

Table 3.5: Genes in Best k-tsp for PGR named by Entrez ID and Hugo Id

3.2 10 Year Recurrence

Accuracies for prediction of recurrence 10 years are shown in table 3.6. Figure 3.7 shows the mean CV performance. The selected genes are shown in table 3.7 p. 62. A detailed table is availble in the appendix table 5.7 p. 85. In table 3.6 note the high voting threshold indicating that the classifier is focusing on one of the classes.

Table 3.6: Performance for prediction of 10 year recurrence risk

	Best k	Threshold	Training Accuracy	Test Accuracy
10 year	22.0	10.0000	0.6806	0.6320

Entrez A	Hugo A	Entrez B	Hugo B
23303	KIF13B	9133	CCNB2
8764	TNFRSF14	10615	SPAG5
8604	SLC25A12	4288	MKI67
330	BIRC3	9787	DLGAP5
10308	ZNF267	7272	TTK
10628	TXNIP	11130	ZWINT
6908	TBP	9319	TRIP13
26292	MYCBP	9833	MELK
5281	PIGF	1033	CDKN3
25972	UNC50	11065	UBE2C
1117	CHI3L2	2305	FOXM1
7405	UVRAG	22974	TPX2
4074	M6PR	891	CCNB1
4285	MIPEP	4751	NEK2
4189	DNAJB9	1062	CENPE
55573	CDV3	890	CCNA2
2791	GNG11	57007	ACKR3
1777	DNASE2	1058	CENPA
6392	SDHD	27338	UBE2S
9183	ZW10	11004	KIF2C
56998	CTNNBIP1	991	CDC20
27095	TRAPPC3	4171	MCM2

Table 3.7: Genes in Best k-tsp for 10 named by Entrez ID and Hugo Id

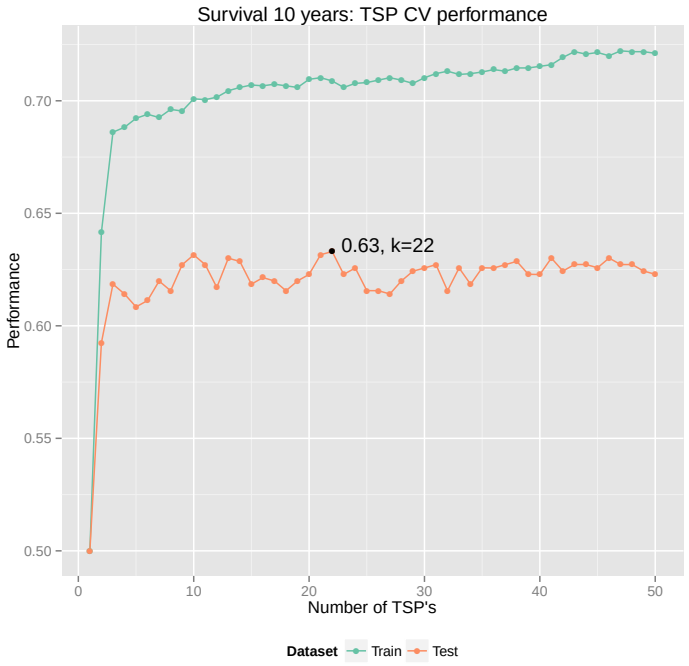

Figure 3.7: Mean performance for 10 year recurrence prediction using 5-fold CV.

3.3 Evaluation of Survival Models

This section presents performance of the different survival models. Table 3.8. For the GP models conditinal plots of all the covariates are included in figures 3.9, 3.10 and 3.11 for the datasets baseline, receptor and fingerprints respectively. All conditional plots are produced using the *notfree* GP because the performance for the *free* and *notfree* models are equal, in which case we prefer the simpler model. The conditional plots are created by tying all but one covariate at their mean value and varying the non-tied parameter.

Table 3.8: Performance of different survival models. In the Freedom column notfree and free refers to the variances being shared or not shared across all dimensions, this setting is only applicable when GP's were used.

Model	Freedom	Train	Test
GP baseline	free	0.6991	0.6775
GP baseline	notfree	0.6984	0.6774
GP receptor	free	0.7086	0.6854
GP receptor	notfree	0.7086	0.6859
GP surv10	free	0.7742	0.6933
GP surv10	notfree	0.7741	0.6934
CoxPH baseline	-	0.6870	0.6812
CoxPH receptor	-	0.6861	0.6834
CoxPH Survival 10 years	-	0.7425	0.6958
NPI	-	0.6492	0.6428
St. Gallen	-	0.5416	0.5533

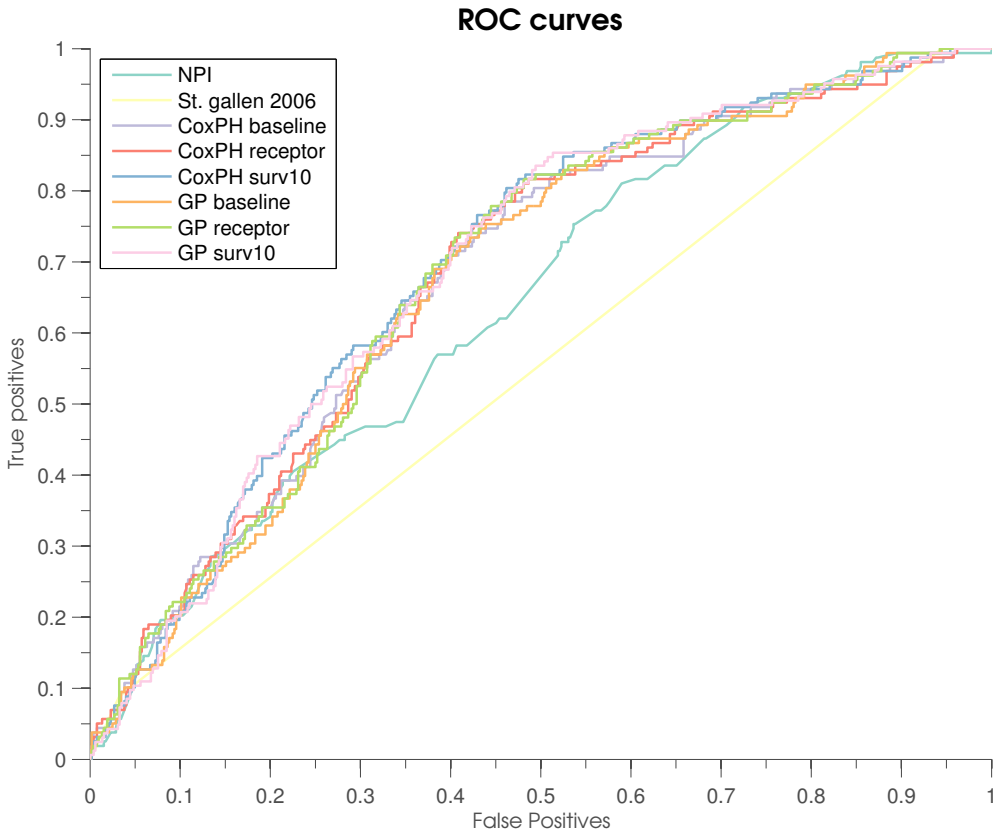

Figure 3.8: ROC curves for survival models

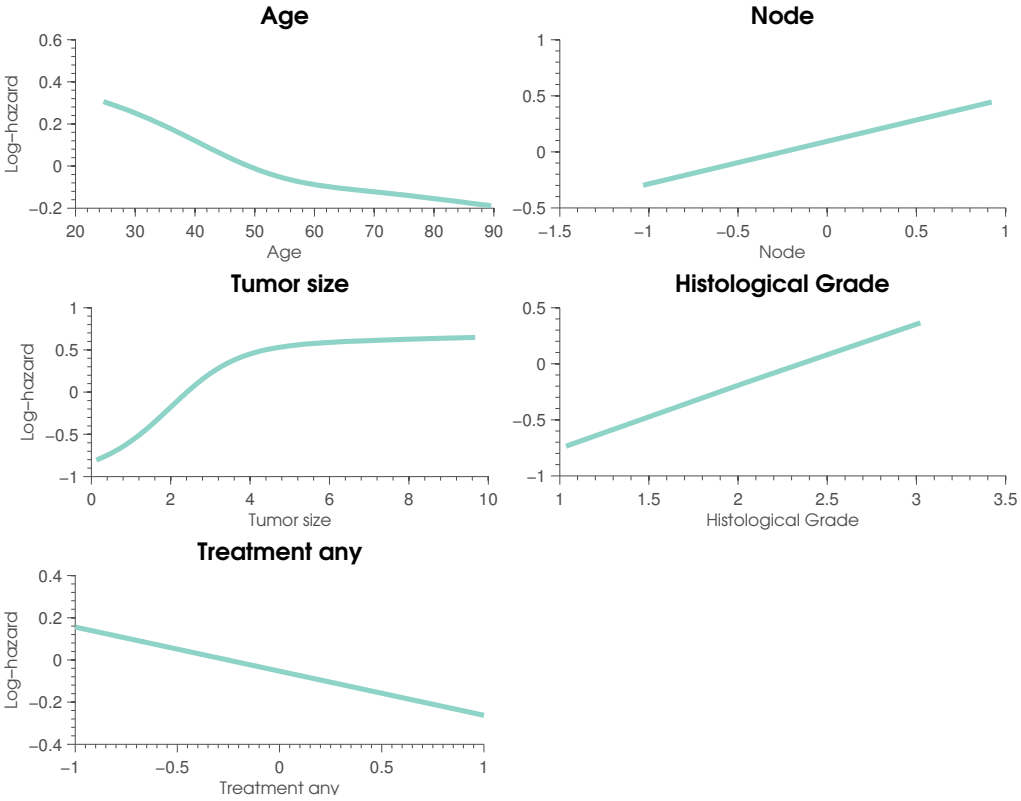

Figure 3.9: Conditional plots for GP model trained using the baseline dataset.

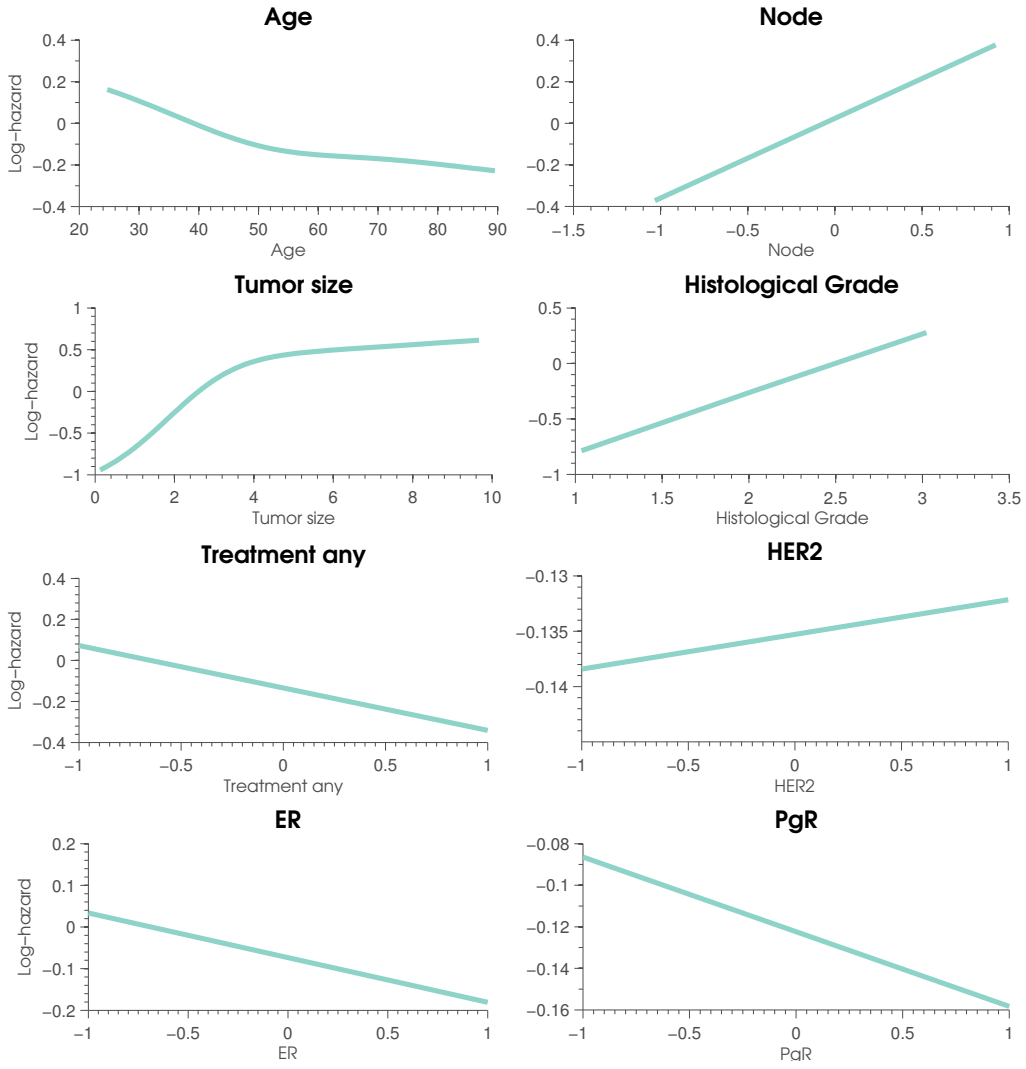

Figure 3.10: Conditional plots for GP model trained using the receptor dataset.

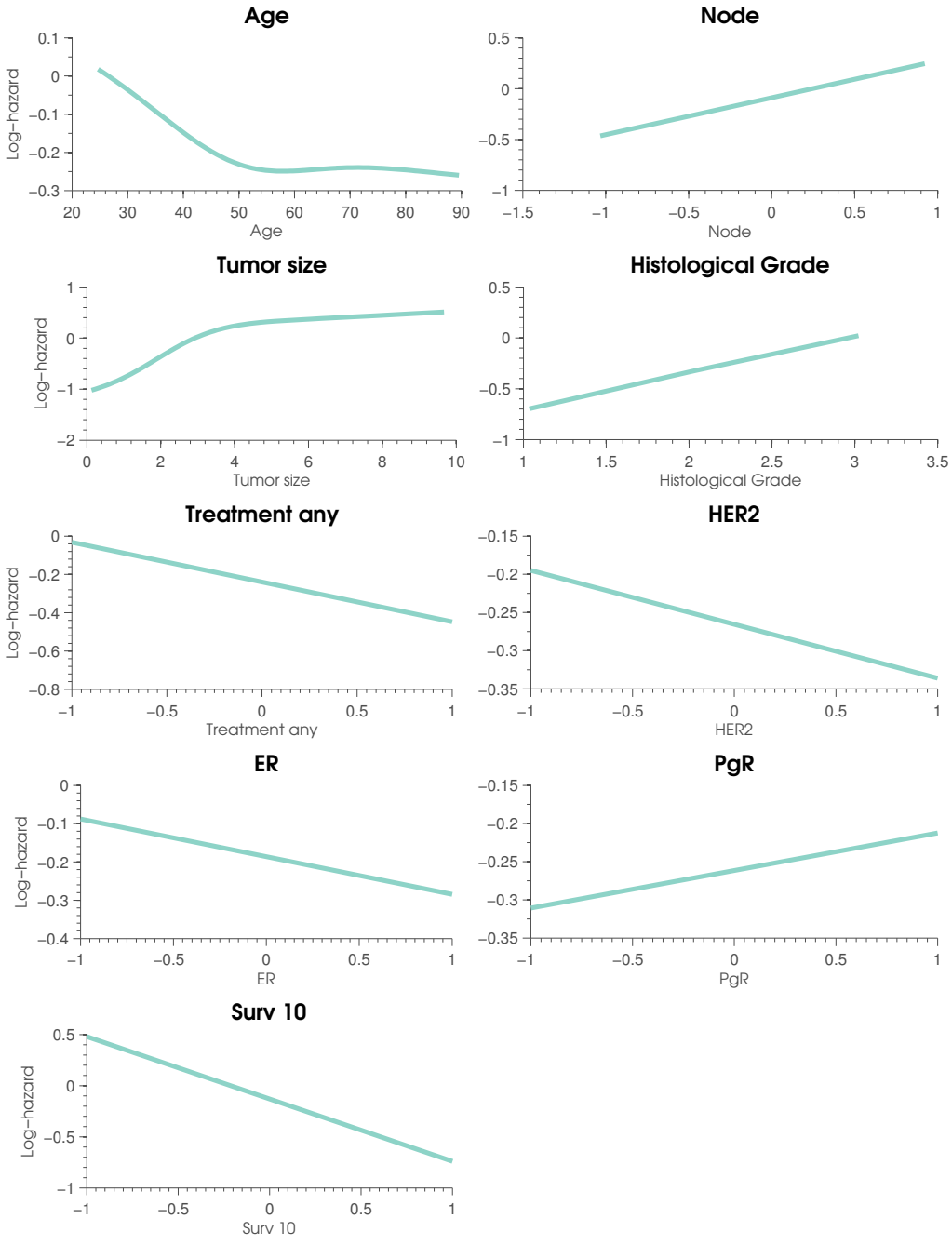

Figure 3.11: Conditional plots for GP model trained using the fingerprint dataset.

Chapter 4

Discussion and Conclusion

The purpose of the project was to evaluate new methods for modeling of survival data, specifically we modeled recurrence risk in breast cancer patients using GP based models, CoxPH models, NPI and STG.

Data

Data was collected from the repository created by Haibe-Kains et al. 2012. The repository contains clinical data and micro array data for breast cancer patients collected in several different studies. The micro array data is measured with several platforms as shown in table 2.4 p. 34. Mapping between platforms was performed by creating a set of entrez ID's which was shared across all platforms. To keep the shared set reasonably large we only considered the platforms: Agilent, Affymetrix HGU A/B, Affymetrix HGU95 and Agilent99. This resulted in 7648 entrez ID's being shard across all platforms. In case of multiple probes mapping to the same ID the highest variance probe, within each study, was preferred.

For the survival models the following clinical covariates were required: age, tumor size, nodal involvement, histological grade and treatment status and recurrence data, 2064 patients had these covariates available. Table 2.5 p. 35 shows statistics on the included patients. Presence or absence of the receptors ER, HER2 and PgR is often used for subtyping of breast cancers. We wanted to investigate the effect of adding receptor status as covariates in the survival models. Table 2.5 show that HER2 and PgR receptor status is

only available for a limited subset of patients (n=278 and 836 respectively). Different methods for inferring receptor status was evaluated, these will be discussed later.

The data from table 2.4 p. 34 was visualized using Kaplan-Meier plots, the plots are presented in section2.9.4 p. 36. As expected the Kaplan-Meier plots show that increased histological grade (p-value: $<$ 2.22e-16), NPI score (p-value: $<$ 2.22e-16), STG score (p-value: 1.7263e-10), tumor size (p-value: 9.992e-16) and nodal involvement (p-value: 1.2167e-07) are associated with increased risk of recurrence. From the Kaplan-Meier plots it seems that NPI and STG are good predictors of recurrence risk, but evaluation of their AUC contradict this. For STG this probably happens because it classifies the majority of the patients in the intermediate group, leaving only the best and worst patients in the good and poor groups, see table 2.6 p. 36. NPI classifies most patients into the groups good and intermediate and seems to be able to discriminate between these, but very few patients are classified in the poor group. NPI and STG are evaluated at prediction on recurrence risk, a task they where not developed for, which is the likely cause of the poor performance. The Kaplan-Meier plot of age shows that the low age group patients has high recurrence risk (p-value: 0.011539). Kaplan-Meier plots of receptor status show that negative ER (p-value: 1.7784e-06) and PgR (p-value: 3.4957e-10) receptor status seems to be associated with increased risk of recurrence. The p-value HER2 is not significant ((p-value: 0.068235). Prediction of recurrence at 10 years (p-value: 0.59559) using micro array data and patients receiving treatment (p-value: 0.72859) does not have significant p-values.

Pilot Study

A pilot study was used to investigate the performance of the survival models NPI, STG, CoxPH, RSF and GP based models. The pilot study evaluated prediction of overall survival as opposed to the full study, which evaluated recurrence free survival. The difference in target between the pilot study and the full study happened because the pilot study only uses clinical data where many patients had overall survival data available. The full study needs clinical and microarray data to be available, the micro array requirement limited the number of available patients making it necessary to use recurrence risk instead of overall survival. The results of the pilot study

are shown in table 2.8 p. 42. The pilot study showed that GP based models generally performed better than both random forest based models and CoxPH based models. Both GP based models and random forest models are computationally demanding to evaluate. Based on the results from the pilot study further investigations focused on the GP based models, CoxPH and NPI and STG, the latter 3 being fast to evaluate.

Inference of Receptors

Receptor status was only available for a subset of the included patients. Current methods for inference of receptor status are developed using the Affymetrix HGU133 platforms and are based on fitting a Gaussian mixture model to the expression value of a single probe. The performance of the Gaussian mixture model method is shown in table 3.1 p. 52. On the dataset used in this study the performance is no better than random for HER2 (45.7%) and PgR (49.3%) and only slightly better than random for ER (59.1%). These results are worse than the results reported by Karn et al. 2010 who used a similar method and obtained accuracies 91.6%, 89.2%, and 71.8% for ER, HER2 and PgR respectively. The method by Karn et al. 2010 uses a single platform, Affymetrix HGU133A, and a specific probe from this platform, to represent each of the receptors. To be able to evaluate the Gaussian mixture model using different platforms we first mapped the probes, representing each receptor to entrez ID's, if several probes mapped to the same entrez ID the probe with maximum variance was used. The probes used to represent each receptor are therefore not the same as the ones used by Karn et al. 2010. The Gaussian mixture approach rely on the expression density having a bimodal shape, in figure 3.1, p. 53 the MSK study is a good example of an entrez ID having a bimodal density. Figure 3.1, 3.2 and 3.3 clearly shows that the densities for many studies are not bimodal. The lack of bimodality can be caused by some studies having only positive or negative samples, in which case the Gaussian mixture does not work, secondly the lack of bimodality may be caused by inclusion of different micro array platforms than Affymetrix HGU133A.

We evaluated an alternative method for inference of receptor status based on relative gene expression. The performance of this method is summarized in table 3.2 p. 56, The test accuracies are 90.01%, 81.93% and 78.19% for ER, HER2 and PgR respectively. The performance using the k-TSP

method is better for predicting PgR (78.2% vs. 71.8%), a little worse for ER (90.0% vs. 91.6%,) and worse for HER2 (81.9% vs. 89.2%). The k-TSP method improves on the method by Karn et al. 2010 by being able to classify single arrays and being robust against normalization and changes in platform. The current implementation of the k-TSP algorithm uses voting among the gene pairs to make a decision. The performance of the k-TSP may be improved by using other methods than voting for combining the predictions of the individual gene pairs, e.g. random forest, SVM's or neural networks.

Table 3.3 p. 58, 3.4 p. 59 and 3.5 p. 61 shows the entrez ID's of the selected gene pairs for ER, HER2 and PgR receptors respectively. The gene pairs are further detailed in appendix tables: 5.4 p. 82, 5.5 p. 83 and 5.6 p. 84 which show gene names of the genes in the gene pairs. Karn et al. 2010 used the probes 205225_at, 216836_s_at and 208305_at to represent ER, HER2 and PgR respectively. These probes corresponds to entrez ID's, ER: 2099, HER2: 2064 and PgR: 5241. For ER entrez 2099 is in the top pair and for HER2 entrez is in gene pair 5. For PgR the entrez ID used by Karn et al. 2010 is not present in the k-TSP.

Micro Array Derived Features

Features derived from micro array data has been shown to be predictive of survival in breast cancer patients. We considered to include PAM50 intrinsic subtypes in the model, but because the algorithm is based on specific probes from Affymetrix chips this was not possible. The same problem was present for OncotypeDX and MammaPrint that used probes that did not map to an entrez ID. To include some micro array derived features we used a k-TSP to predict recurrence risk at 10 years represented as a binary indicator. The results are shown in table 3.6 p. 62. For predicting 10 years recurrence the training performance is 68.1% and the test accuracy is 63.2% indicating that the method is overfitting. The k-TSP uses 22 gene pairs and a voting threshold value of 10.

Survival Models

The models CoxPH, NPI, STG and GP based survival models were evaluated for their ability for predicting 10-year recurrence in breast cancer

patients. The results are shown in table 3.8 p. 64 and are visualized in figure 3.8 p. 65. Table 3.8 shows that the performance for GP models and CoxPH models is similar. The NPI and STG models are not effective for predicting recurrence.

Table 3.8 shows that addition of additional features increase the performance for both GP based models and CoxPH models. Addition of the micro array based prediction of 10 year recurrence increase the training performance more than 5% for both GP models and CoxPH models. The increases in test performance are modest indicating that both models might be overfitting.

In GP models the degree of freedom, *free* and *notfree* in table 3.8, does not impact performance, in which case we should prefer the *notfree* because it has less free parameters. Conditional plots for GP *notfree* models are shown in figures 3.9, 3.10 and 3.11 for datasets baseline, receptor and fingerprints respectively. The conditional plots show the effect of varying a single covariate while the other covariates are held constant. The conditional plots shows that nodal involvement, tumor size and histological grade are the strongest predictors of recurrence. Interestingly age is associated with decreased risk of recurrence, which is the opposite effect of what is seen in the pilot study. This change is probably due to the change from overall to recurrence risk as outcome variable. Receiving no treatment is associated with increased recurrence risk. The conditional plots also shows that ER negative status (-1 in the conditional plots) is associated with increased risk. For PgR the conditional plots are nearly constant performance and the datasets receptor and fingerprint does not agree whether PgR negative status is associated with decreased or increased recurrence risk. In the fingerprint data, figure 3.11, predicted recurrence at 10 years is associated with increased risk. For the surv10 plot -1 is predicted recurrence and 1 is right censoring.

We conclude that addition of both receptor status and micro array derived features improve the predictive performance for both GP models and CoxPH models. GP based models are not better than CoxPH models for prediction of recurrence in breast cancer patients.

Bibliography

Breiman, Leo (2001). *Random forests*. Vol. 45. Springer.

Chambless, Lloyd E, Christopher P Cummiskey, and Gang Cui (Jan. 2011). "Several methods to assess improvement in risk prediction models: extension to survival analysis." In: *Statistics in medicine* 30.1, pp. 22–38.

Cox, David R (1972). "Regression models and life-tables". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 187–220.

Curtis, Christina et al. (June 2012). "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups." In: *Nature* 486.7403, pp. 346–352.

Desantis, Carol et al. (Oct. 2013). "Breast cancer statistics, 2013." In: *CA: a cancer journal for clinicians*.

Eddy, James A et al. (Apr. 2010). "Relative expression analysis for molecular cancer diagnosis and prognosis." In: *Technol Cancer Res Treat* 9.2, pp. 149–159.

Galea, M H et al. (1992). "The Nottingham Prognostic Index in primary breast cancer." In: *Breast cancer research and treatment* 22.3, pp. 207–219.

Geman, Donald et al. (2004). "Classifying gene expression profiles from pairwise mRNA comparisons." In: *Statistical applications in genetics and molecular biology* 3, Article19.

Goldhirsch, A et al. (Sept. 2013). "Personalizing the treatment of women with early breast cancer: highlights of the St Gallen International Expert Consensus on the Primary Therapy of Early Breast Cancer 2013." In:

Annals of oncology : official journal of the European Society for Medical Oncology / ESMO 24.9, pp. 2206–2223.

Goldhirsch, Aron et al. (Sept. 2003). "Meeting highlights: updated international expert consensus on the primary therapy of early breast cancer." In: *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*. International Breast Cancer Study Group, Oncology Institute of Southern Switzerland, Lugano. agoldhirsch@sakk.ch., pp. 3357–3365.

Haibe-Kains, B et al. (Feb. 2012). "A Three-Gene Model to Robustly Identify Breast Cancer Molecular Subtypes". In: *JNCI Journal of the National Cancer Institute* 104.4, pp. 311–325.

Haibe-Kains, Benjamin et al. (2013). *genefu: Relevant Functions for Gene Expression Analysis, Especially in Breast Cancer*. R package version 1.11.0. URL: `http://compbio.dfci.harvard.edu`.

Hastie, Trevor et al. *impute: impute: Imputation for microarray data*. R package version 1.35.0.

Hastie, Trevor et al. (1999). *Imputing missing data for gene expression arrays*.

Ibrahim, Joseph G, Ming-Hui Chen, and Debajyoti Sinha (June 2001). *Bayesian Survival Analysis*. Springer.

Ishwaran, H. and U.B. Kogalur (2013). "Random Forests for Survival, Regression and Classification(RF-SRC)". In: R package version 1.3. URL: `http://cran.r-project.org/web/packages/randomForestSRC/`.

Ishwaran, Hemant et al. (Nov. 2008). "Random survival forests". In: *arXiv.org* stat.AP.

Joensuu, Heikki et al. (Mar. 2012). "Risk of recurrence of gastrointestinal stromal tumour after surgery: an analysis of pooled population-based cohorts." In: *The lancet oncology* 13.3, pp. 265–274.

Karn, Thomas et al. (Apr. 2010). "Data-driven derivation of cutoffs from a pool of 3,030 Affymetrix arrays to stratify distinct clinical types of breast cancer." In: *Breast cancer research and treatment* 120.3, pp. 567–579.

Leek, Jeffrey T (May 2009). "The tspair package for finding top scoring pair classifiers in R." In: *Bioinformatics* 25.9, pp. 1203–1204.

Lehmann, Brian D et al. (July 2011). "Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies." In: *Journal of Clinical Investigation* 121.7, pp. 2750–2767.

Leisch, Friedrich (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis". In: *Compstat 2002 — Proceedings in Computational Statistics*. Ed. by Wolfgang Härdle and Bernd Rönz. ISBN 3-7908-1517-9. Physica Verlag, Heidelberg, pp. 575–580. URL: `http://www.stat.uni-muenchen.de/~leisch/Sweave`.

Lin, Xue et al. (2009). "The ordering of expression among a few genes can provide simple cancer biomarkers and signal BRCA1 mutations." In: *BMC Bioinformatics* 10, p. 256.

Magis, Andrew T et al. (Mar. 2011). "Graphics processing unit implementations of relative expression analysis algorithms enable dramatic computational speedup." In: *Bioinformatics* 27.6, pp. 872–873.

Marchionni, Luigi et al. (Mar. 2008). "Systematic review: gene expression profiling assays in early-stage breast cancer." In: *Annals of internal medicine* 148.5, pp. 358–369.

Marchionni, Luigi et al. (2013). "A simple and reproducible breast cancer prognostic test". In: *BMC Genomics* 14.1, p. 336.

Paik, Soonmyung et al. (Dec. 2004). "A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer." In: *New England Journal of Medicine* 351.27, pp. 2817–2826.

Parker, Joel S et al. (Mar. 2009). "Supervised risk predictor of breast cancer based on intrinsic subtypes." In: *Journal of clinical oncology : official journal of the American Society of Clinical Oncology* 27.8, pp. 1160–1167.

Rasmussen, C. E. and K. I. Williams (2006). "Gaussian Processes for Machine Learning ". In: Massachusetts: MIT Press.

Sørlie, T et al. (Sept. 2001). "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications." In: *Proceedings of the National Academy of Sciences of the United States of America* 98.19, pp. 10869–10874.

Tan, Aik Choon et al. (Oct. 2005). "Simple decision rules for classifying human cancers from gene expression profiles." In: *Bioinformatics* 21.20, pp. 3896–3904.

Vanhatalo, Jarno et al. (2013). "GPstuff: Bayesian modeling with Gaussian processes". In: *The Journal of Machine Learning Research* 14.1, pp. 1175–1179.

Veer, Laura J van 't et al. (Jan. 2002). "Gene expression profiling predicts clinical outcome of breast cancer." In: *Nature* 415.6871, pp. 530–536.

Vijver, Marc J van de et al. (Dec. 2002). "A gene-expression signature as a predictor of survival in breast cancer." In: *New England Journal of Medicine* 347.25, pp. 1999–2009.

Zeileis, Achim et al. (2008). "Implementing a class of permutation tests: The coin package". In: *Journal of Statistical Software* 28.8, pp. 1–23.

Chapter 5

Appendix

5.1 Receptor inference Gaussian mixture datasets

study	platform	pos	neg	frac
SUPERTAM_HGU133PLUS2	affy	164	0	1.00
SUPERTAM_HGU133A	affy	507	5	0.99
UPP	affy	213	34	0.86
KOO	affy.u95	73	15	0.83
STK	affy	130	29	0.82
MAINZ	affy	162	38	0.81
NKI	agilent	249	88	0.74
UNT	affy	86	40	0.68
TRANSBIG	affy	134	64	0.68
DUKE	affy.u95	114	57	0.67
EXPO	affy	161	85	0.65
CAL	affy	75	43	0.64
MDA4	affy	79	48	0.62
MAQC2	affy	141	89	0.61
DFHCC	affy	70	45	0.61
UNC4	agilent99	154	99	0.61
VDX	affy	209	135	0.61
IRB	affy	76	53	0.59
MSK	affy	57	42	0.58
EORTC10994	affy	27	22	0.55
PNC	affy	45	43	0.51
DFHCC2	affy	31	53	0.37

Table 5.1: ER receptor. Datasets and distribution of negative and postive ER receptors. Frac is the fraction of ER positives.

study	platform	pos	neg	frac
MSK	affy	85	0	1.00
DFHCC	affy	36	79	0.31
PNC	affy	26	64	0.29
EXPO	affy	61	166	0.27
IRB	affy	31	98	0.24
UNC4	agilent99	58	203	0.22
DFHCC2	affy	18	66	0.21
MAQC2	affy	40	190	0.17
MDA4	affy	15	114	0.12

Table 5.2: HER2 receptor. Datasets and distribution of negative and postive HER2 receptors. Frac is the fraction of HER2 positives.

study	platform	pos	neg	frac
SUPERTAM_HGU133A	affy	65	5	0.93
UNT	affy	56	6	0.90
SUPERTAM_HGU133PLUS2	affy	123	39	0.76
UPP	affy	190	61	0.76
KOO	affy.u95	65	23	0.74
DUKE	affy.u95	65	23	0.74
CAL	affy	66	51	0.56
DFHCC	affy	64	51	0.56
EXPO	affy	129	114	0.53
PNC	affy	40	43	0.48
UNC4	agilent99	109	126	0.46
MAQC2	affy	104	126	0.45
MSK	affy	43	55	0.44
MDA4	affy	54	73	0.43
EORTC10994	affy	18	29	0.38
DFHCC2	affy	31	53	0.37

Table 5.3: PgR receptor. Datasets and distribution of negative and postive PgR receptors. Frac is the fraction of PgR positives

5.2 Receptor inference gene names

Gene Pair	Entrez A	Gene Name	Entrez B	Gene Name B
1	2099	estrogen receptor 1	4953	ornithine decarboxylase 1
2	2625	GATA binding protein 3	8884	solute carrier family 5 (sodium/multivitamin and iodide cotransporter), member 6
3	23158	TBC1 domain family, member 9 (with GRAM domain)	114908	transmembrane protein 123
4	771	carbonic anhydrase XII	5214	phosphofructokinase, platelet
5	7802	dynein, axonemal, light intermediate chain 1	6491	SCL/TAL1 interrupting locus
6	9	N-acetyltransferase 1 (arylamine N-acetyltransferase)	9111	N-myc (and STAT) interactor
7	18	4-aminobutyrate aminotransferase	10926	DBF4 homolog (S. cerevisiae)
8	4602	v-myb avian myeloblastosis viral oncogene homolog	2195	FAT atypical cadherin 1
9	7033	trefoil factor 3 (intestinal)	6590	secretory leukocyte peptidase inhibitor
10	7031	trefoil factor 1	991	cell division cycle 20
11	8416	annexin A9	9212	aurora kinase B
12	2066	v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 4	53335	B-cell CLL/lymphoma 11A (zinc finger protein)
13	4137	microtubule-associated protein tau	898	cyclin E1
14	10551	anterior gradient 2	6280	S100 calcium binding protein A9
15	9687	growth regulation by estrogen in breast cancer 1	6664	SRY (sex determining region Y)-box 11
16	8614	stanniocalcin 2	1001	cadherin 3, type 1, P-cadherin (placental)
17	2203	fructose-1,6-bisphosphatase 1	8140	solute carrier family 7 (amino acid transporter light chain, L system), member 5
18	7494	X-box binding protein 1	7388	ubiquinol-cytochrome c reductase hinge protein
19	51097	saccharopine dehydrogenase (putative)	8833	guanine monphosphate synthase
20	1602	dachshund homolog 1 (Drosophila)	51442	vestigial like 1 (Drosophila)
21	8382	NME/NM23 family member 5	1058	centromere protein A
22	10974	adipogenesis regulatory factor	445	argininosuccinate synthase 1
23	6337	sodium channel, non-voltage-gated 1 alpha subunit	5918	retinoic acid receptor responder (tazarotene induced) 1
24	22977	aldo-keto reductase family 7, member A3 (aflatoxin aldehyde reductase)	11004	kinesin family member 2C
25	3480	insulin-like growth factor 1 receptor	2296	forkhead box C1
26	2674	GDNF family receptor alpha 1	9420	cytochrome P450, family 7, subfamily B, polypeptide 1
27	2879	glutathione peroxidase 4	4904	Y box binding protein 1
28	3249	hepsin	1054	CCAAT/enhancer binding protein (C/EBP), gamma
29	79921	transcription elongation factor A (SII)-like 4	7913	DEK oncogene
30	6478	siah E3 ubiquitin protein ligase 2	9833	maternal embryonic leucine zipper kinase

Table 5.4: Genes in Best k-tsp for ER named by Entrez ID and gene names.

Gene Pair	Entrez A	Gene Name	Entrez B	Gene Name B
1	10948	StAR-related lipid transfer (START) domain containing 3	596	B-cell CLL/lymphoma 2
2	2886	growth factor receptor-bound protein 7	1108	chromodomain helicase DNA binding protein 4
3	51755	cyclin-dependent kinase 12	23131	G patch domain containing 8
4	2064	v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2	6659	SRY (sex determining region Y)-box 4
5	5709	proteasome (prosome, macropain) 26S subunit, non-ATPase, 3	6651	SON DNA binding protein
6	8564	kynurenine 3-monooxygenase (kynurenine 3-hydroxylase)	9639	Rho guanine nucleotide exchange factor (GEF) 10
7	5409	phenylethanolamine N-methyltransferase	2145	enhancer of zeste homolog 1 (Drosophila)
8	8714	ATP-binding cassette, sub-family C (CFTR/MRP), member 3	23189	KN motif and ankyrin repeat domains 1

Table 5.5: Genes in Best k-tsp for HER2 named by Entrez ID and gene names.

Gene Pair	Entrez A	Gene Name	Entrez B	Gene Name B
1	771	carbonic anhydrase XII	4860	purine nucleoside phosphorylase
2	2099	estrogen receptor 1	4704	NADH dehydrogenase (ubiquinone) 1 alpha subcomplex, 9, 39kDa
3	2625	GATA binding protein 3	11130	ZW10 interacting kinetochore protein
4	7802	dynein, axonemal, light intermediate chain 1	11339	Opa interacting protein 5
5	9687	growth regulation by estrogen in breast cancer 1	7272	TTK protein kinase
6	9	N-acetyltransferase 1 (arylamine N-acetyltransferase)	1475	cystatin A (stefin A)
7	18	4-aminobutyrate aminotransferase	9833	maternal embryonic leucine zipper kinase
8	2066	v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 4	1054	CCAAT/enhancer binding protein (C/EBP), gamma
9	4137	microtubule-associated protein tau	898	cyclin E1

Table 5.6: Genes in Best k-tsp for PGR named by Entrez ID and gene names.

Gene Pair	Entrez A	Gene Name	Entrez B	Gene Name B
1	23303	kinesin family member 13B	9133	cyclin B2
2	8764	tumor necrosis factor receptor superfamily, member 14	10615	sperm associated antigen 5
3	8604	solute carrier family 25 (aspartate/glutamate carrier), member 12	4288	antigen identified by monoclonal antibody Ki-67
4	330	baculoviral IAP repeat containing 3	9787	discs, large (Drosophila) homolog-associated protein 5
5	10308	zinc finger protein 267	7272	TTK protein kinase
6	10628	thioredoxin interacting protein	11130	ZW10 interacting kinetochore protein
7	6908	TATA box binding protein	9319	thyroid hormone receptor interactor 13
8	26292	MYC binding protein	9833	maternal embryonic leucine zipper kinase
9	5281	phosphatidylinositol glycan anchor biosynthesis, class F	1033	cyclin-dependent kinase inhibitor 3
10	25972	unc-50 homolog (C. elegans)	11065	ubiquitin-conjugating enzyme E2C
11	1117	chitinase 3-like 2	2305	forkhead box M1
12	7405	UV radiation resistance associated	22974	TPX2, microtubule-associated
13	4074	mannose-6-phosphate receptor (cation dependent)	891	cyclin B1
14	4285	mitochondrial intermediate peptidase	4751	NIMA-related kinase 2
15	4189	DnaJ (Hsp40) homolog, subfamily B, member 9	1062	centromere protein E, 312kDa
16	55573	CDV3 homolog (mouse)	890	cyclin A2
17	2791	guanine nucleotide binding protein (G protein), gamma 11	57007	atypical chemokine receptor 3
18	1777	deoxyribonuclease II, lysosomal	1058	centromere protein A
19	6392	succinate dehydrogenase complex, subunit D, integral membrane protein	27338	ubiquitin-conjugating enzyme E2S
20	9183	zw10 kinetochore protein	11004	kinesin family member 2C
21	56998	catenin, beta interacting protein 1	991	cell division cycle 20
22	27095	trafficking protein particle complex 3	4171	minichromosome maintenance complex component 2

Table 5.7: Genes in Best k-tsp for 10 YEAR RECURRENCE named by Entrez ID and gene names.

5.3 Gaussian Processes

A Gaussian process is defined as a probability distribution over functions f(x) such that f(x) evaluated at any set of points $x_1 ... x_n$ is jointly Gaussian.

We define the Gaussian process, GP with:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{5.1}$$

$$\text{Mean} \quad m(x) = E[f(x)] \tag{5.2}$$

$$\text{Covariance} \quad k(x, x') = E[(f(x).m(x))(f(x') - m(x'))] \tag{5.3}$$

$$X \in \mathfrak{Re}^D \tag{5.4}$$

We often assumes that $m(x) = 0$. $f(x)$ is the process evaluated at the point $\mathbf{x}$ Using the definition of a GP we can draw a number of functions, $\mathbf{f}_*$ from a particular GP evaluated at the points X_*, i.e:

$$\mathbf{f}_* \sim \mathcal{N}(0, K(X_*, X_*)) \tag{5.5}$$

The following is a short demonstration of 5.5. In order to draw functions from the $\mathcal{GP}$ we need to defines its covariance functions. This example uses the squared exponential covariance function:

$$k(x, x') = \sigma_f \cdot \exp\left(\frac{-(x - x')^2}{2 \cdot l^2}\right) \tag{5.6}$$

Here σ_f is magnitude parameter governing the overall variability of the process and l is the length scale governing the governing the correlation between points. Both σ_f and l hyper parameters of the GP[1]. To draw functions from the GP do:

1. Create a vector of test inputs, X_* at which to evaluate the function
2. calculate the covariance matrix, $K(X_*, X_*)$ with the covariance function $k(x, x')$, e.g with equation 5.6 or some other valid kernel function. See figure 5.1 panel C) for example covariance function.

[1] `http://skaae.shinyapps.io/test_project/` lets you play with the hyper parameters. The example is also available at `https://github.com/skaae/GP_shiny/` with instruction on how to run the example locally

3. Draw multivariate samples from multivariate function

   $\mathbf{f}_* \sim \mathcal{N}(0, K(X_*, X_*))$, where $\mathbf{f}_*$ is evaluation of the GP at the test points X_*

The code in listing 5.1 and listing 5.2 was used to draw functions from a GP. All code in the listings is available at `https://bitbucket.org/skaae/simple_gp_matlab`. The plots from listing 5.1 is shown in figure 5.1. Panel A) shows 3 functions drawn from the GP, panel B) shows the histogram of $x = -3.1633$ evaluated 5000 times, the plot shows that f(x) is Gaussian. Lastly panel C) shows the covariance matrix.

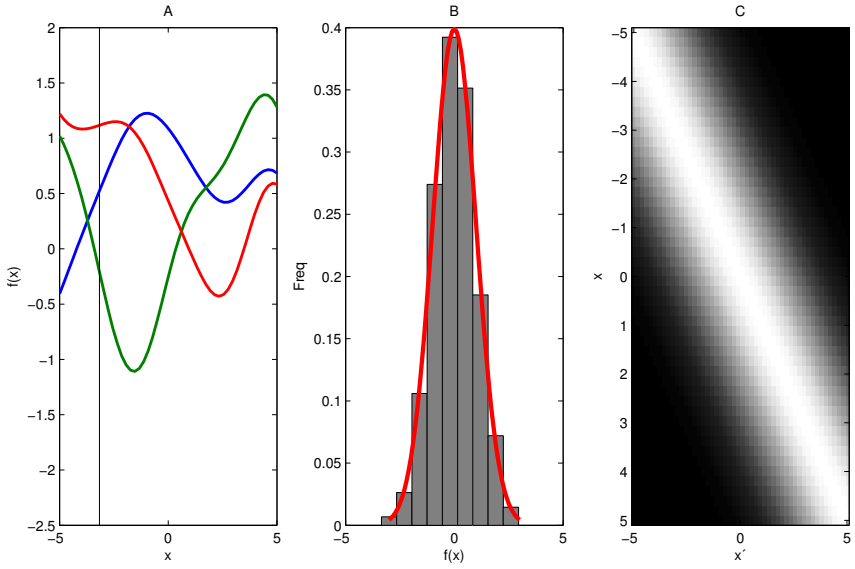

Figure 5.1: Panel A) shows 3 functions drawn from the GP, panel B) shows the histogram of $x = -3.1633$ evaluated 5000 times, the plot shows that f(x) is Gaussian. Panel C) shows the covariance matrix.

Listing 5.1: Draw functions from GP

```
rand('state',12345);
x_star = linspace(-5,5,50);  % test data
sigma_f= 1; l = 1; f_samples = 5000;, sigma_n = 0.5;
```

```
% squarred exp kernel
sqr_exp = @(x1,x2) sigma_f * exp(-(x1-x2)^2 / 2*l^2);
kernel = sqr_exp
K = calc_k(x_star,x_star,0,sqr_exp);
gp_samples = mvnrnd(zeros(1,length(K)), K, f_samples);

%% plots
subplot(1,3,1); plot(repmat(x_star',1,3),...
                     gp_samples(1:3,:)','LineWidth',2)
hold on; plot([x_star(10) x_star(10)],[2,-2.5],'k-'); hold off;
ylabel('f(x)'),xlabel('x'),title('A')

subplot(1,3,2);
binWidth = 0.7; %This is the bin width
binCtrs = -3:binWidth:3; %Bin centers, depends on your data
n=length(gp_samples(:,10));
counts = hist(gp_samples(:,10),binCtrs);
prob = counts / (n * binWidth);
H = bar(binCtrs,prob,'hist');
set(H,'facecolor',[0.5 0.5 0.5]); hold on;
gaus = normpdf(-3:.1:3,0,K(10,10)); %requires Statistics toolbox
plot(-3:.1:3,gaus,'r','linewidth',3);
xlabel('f(x)'); ylabel('Freq'),title('B')
hold off;

subplot(1,3,3); imagesc(x_star,x_star,K/max(max(K)));
colormap('gray');
ylabel('x'),xlabel('xp'),title('C');
```

Listing 5.2: Function for calculation covariance matrix

```
function [ K ] = calc_k(x1,x2,noise,kernel )
% Calculate covariance
K = zeros(length(x1),length(x2));
I = eye(size(K));
for i = 1:size(K,1)
    for j = 1:size(K,2)
        K(i,j) = kernel(x1(i),x2(j));
    end
end
K = K + I .* noise ;
end
```

5.3.1 GP regression

In the regression setting we are interested in drawing functions from the GP. First step in the inference step is to condition the functions drawn from the GP on the training data $\mathcal{D} = \{(x_1, y_1), ...m(x_n, y_n)\}$, where X is the training

input and $\mathbf{y}$ is the training output. We assume that our observations of the targets y are corrupted by noise, i.e. the observation model is assumed to be:

$$\mathbf{y} = \mathbf{f} + \epsilon, \epsilon \sim \mathcal{N}(0, I\sigma_n^2) \tag{5.7}$$

Because the noise is assumed to be independent of $\mathbf{f}$ we use that sum of independent Gaussian the sum of the means and the sum of the covariance, i.e.

$$\mathbf{y} \sim \mathcal{N}(0, K(X,X) + I\sigma_n^2) \tag{5.8}$$
$$\tag{5.9}$$

Following the notation of Rasmussen and Williams 2006, we have that the joint distribution between our observed targets, $\mathbf{y}$ and the points that we want to evaluate $\mathbf{f}_*$ is given by:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{5.10}$$

To get the distribution of $\mathbf{f}_*$ we use standard rules of conditioning [2] on multivariate Gaussians, which gives the distribution of functions conditioned on the test data (X_*), training data (X) and the observed targets ($\mathbf{y}$):

$$\mathbf{f}_*|X,\mathbf{y},X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, cov(\mathbf{f}_*)) \tag{5.11}$$
$$\bar{\mathbf{f}}_* = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} \tag{5.12}$$
$$cov(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(X,X_*) \tag{5.13}$$

The equations above are used in listing 5.3 where we first condition on noise free data and then on data including noise.

Figure 5.2 shows the figure produced by listing 5.3. Note than in panel A) all functions passes through the observed data because we have assumed that the observations has no noise. In panel B) noise is added and the functions are allowed not to move through the observations. Panel C) shows the mean of $\mathbf{f}_*|X,\mathbf{y},X_*$ where the shade indicate uncertainty.

[2] See Rasmussen and Williams 2006 appendix A.2, p. 200, `http://www.gaussianprocess.org/gpml/chapters/RWA.pdf`

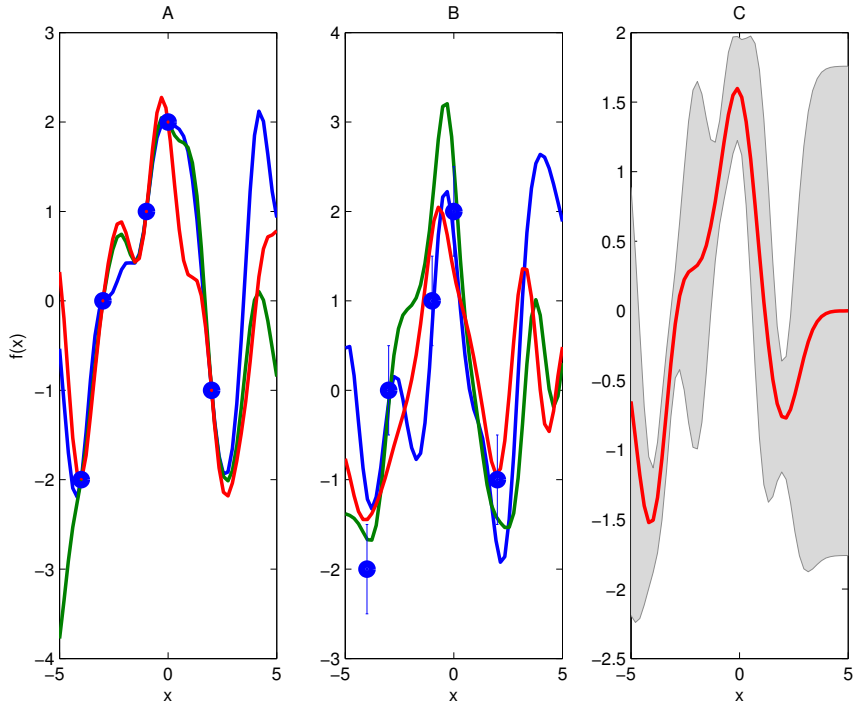

Figure 5.2: A) $\mathbf{f}_*|X,\mathbf{y},X_*$ using noise free observations, B) shows $\mathbf{f}_*|X,\mathbf{y},X_*$ using observations with nose σ_n^2, and C) shows $\bar{\mathbf{f}}_*$ using observations with noise. In C) The shaded error is the uncertainty.

Listing 5.3: conditioning f on data

```
%% samples conditioned on observed data
% calculate mean and covariance of training data (2.23 and 2.24)
% calculate k see Rasmussen (2.21)
% Observed data
x          = [-4,-3,-1,0,2];
y          = [-2,0,1,2,-1]';

k_xx_nonoise      = calc_k(x,x,0,kernel);
k_xx      = calc_k(x,x,sigma_n,kernel);
k_xxs     = calc_k(x,x_star,0,kernel);
k_xsx     = calc_k(x_star,x,0,kernel);
k_xsxs    = calc_k(x_star,x_star,0,kernel);

f_mean_s = k_xsx * inv(k_xx_nonoise) * y;
f_cov_s  = k_xsxs - k_xsx * inv(k_xx_nonoise) * k_xxs;
```

```
gp_samples_wdata_nonoise = mvnrnd(f_mean_s,f_cov_s,3);

f_mean_s = k_xsx * inv(k_xx) * y;
f_cov_s  = k_xsxs - k_xsx * inv(k_xx) * k_xxs;
gp_samples_wdata = mvnrnd(f_mean_s,f_cov_s,3);

figure(2)
subplot(1,3,1); plot(repmat(x_star',1,3),gp_samples_wdata_nonoise',...
    ...
                    'LineWidth',2);
ylabel('f(x)'), xlabel('x'); title('A')
hold on; plot(x,y,'bo','LineWidth',5); hold off;
subplot(1,3,2); plot(repmat(x_star',1,3),gp_samples_wdata',...
                     'LineWidth',2)

hold on; errorbar(x,y,ones(1,length(x)).*sigma_n,'.');
plot(x,y,'bo','LineWidth',5); hold off;
xlabel('x'); title('B')

subplot(1,3,3); shadedErrorBar(x_star,f_mean_s,diag(f_cov_s)); hold...
    on;
plot(x_star,f_mean_s,'r','LineWidth',2); hold off;
xlabel('x'); title('C')
```

5.3.2 Tuning the hyper parameters

We tune the hyper parameters by optimizing the marginal likelihood. The marginal likelihood (evidence) is defined as:

$$\text{Marginal Likelihood: } p(\mathbf{y}|X) = \int \text{likelihood} \times \text{function prior} \tag{5.14}$$

$$= \int p(\mathbf{y}|\mathbf{f},X)p(\mathbf{f}|X)d\mathbf{f}$$

$$\mathbf{f} \quad : \text{training output} \tag{5.15}$$

$$\mathbf{f}_* \quad : \text{test output} \tag{5.16}$$

Where marginal refers to marginalization over the function values $\mathbf{f}$. Equation 5.1 shows the definition of a GP, assuming that the mean is 0 we get the prior distribution:

$$p(\mathbf{f}|X) \sim \mathcal{N}(0, K(X,X)) \tag{5.17}$$

The likelihood is the probability of the targets given training out put:

$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, I\sigma_n^2 \tag{5.18}$$

The likelihood function is conditioned on the training output. If we condition on the training outputs the distribution of $\mathbf{y}$ will be Guassian with mean $\mathbf{f}$ and covariance $I\sigma_n^2$. To optimize this we find the log marginal likelihood as:

$$\begin{aligned} \mathbf{y} \sim \mathcal{N}(0, K(X,X) + \sigma_n^2 I) &= 2\pi^{-\frac{n}{2}} \left|K + \sigma_n^2\right|^{-\frac{1}{2}} \\ &\cdot \exp\left(-\frac{1}{2}\mathbf{y}^T \left(K + \sigma_n^2\right)\mathbf{y}\right) \end{aligned} \tag{5.19}$$

$$\begin{aligned} \ln(l) &= -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|K + \sigma_n^2 I| \\ &- \frac{1}{2}\mathbf{y}^T \left(K + \sigma_n^2 I\right)\mathbf{y} \end{aligned} \tag{5.20}$$

Listing 5.4 shows how the hyper parameters can be found using grid search. In the example the observation noise σ_n^2 is keept at 0.1 and the maximum marginal likelihood is found. The result of running listing 5.4 is shown in figure 5.3. The left panel shows a contour plot of the negative log-likelihood, where the red dot shows the minimum. The middle plot shows three functions drawn from the function posterior and the right plot shows uncertainty and mean prediction.

Listing 5.4: conditioning f on data

```
%% varying the hyper parameters
s_n = 0.1;
resolution = 200;
s_f = linspace(0,5,resolution);
l_f = linspace(0.00000,2,resolution);

marg_loglik = @(n,K,y,s_n) -0.5*n*log(2*pi)...
-0.5*log( det( K+eye(length(K))*s_n) ) ...
-0.5*y'* inv(K+eye(length(K))*s_n)*y;
grid_mll = zeros(resolution);
for i = 1:length(s_f)
    for j = 1:length(l_f)
      sqr_exp = @(x1,x2) s_f(i) * exp(-(x1-x2)^2 / 2*l_f(j)^2);
      K = calc_k(x,x,s_n,sqr_exp);
      n = length(y);
      grid_mll(i,j) = marg_loglik(n,K,y,s_n);
    end
end
fprintf('\n');

% plot negative loglikeihood, normalized
% we need to find the minimum
```

```
figure(3); subplot(1,3,1);
im = -grid_mll ./ max(max(-grid_mll));
contour(l_f,s_f,im);
ylabel('length'); xlabel('\sigma_f^2'); title('negative normalized log lik')
[v,ind]=min(im(:));
[sf_minidx,lf_minidx] = ind2sub(size(im),ind);
lf_min = l_f(lf_minidx);
sf_min = s_f(sf_minidx);
hold on;
plot(lf_min,sf_min,'or','LineWidth',6);
hold off;

kernel = @(x1,x2) sf_min * exp(-(x1-x2)^2 / 2*lf_min^2);
k_xx     = calc_k(x,x,s_n,kernel);
k_xxs    = calc_k(x,x_star,0,kernel);
k_xsx    = calc_k(x_star,x,0,kernel);
k_xsxs   = calc_k(x_star,x_star,0,kernel);

f_mean_s = k_xsx * inv(k_xx) * y;
f_cov_s  = k_xsxs - k_xsx * inv(k_xx) * k_xxs;
gp_samples_wdata = mvnrnd(f_mean_s,f_cov_s,3);

subplot(1,3,2); plot(repmat(x_star',1,3),gp_samples_wdata',...
                     'LineWidth',2)
hold on; errorbar(x,y,ones(1,length(x)).*s_n,'.'); hold off;
xlabel('x'); title('3 functions drawn from Posterior')

subplot(1,3,3); shadedErrorBar(x_star,f_mean_s,diag(f_cov_s)); hold on;
plot(x_star,f_mean_s,'r','LineWidth',2);
plot(x,y,'b.')
hold off;
xlabel('x'); title('Uncertainty and mean')
```

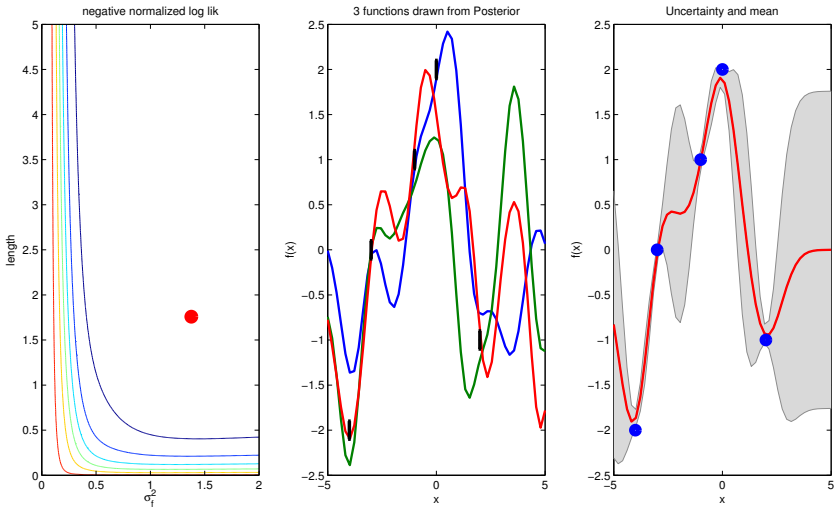

Figure 5.3: The left panel shows a contour plot of the negative loglikelihood, where the red dot shows the minimum. The middle plot shows three functions drawn from the function posterior and the right plot shows uncertainty and mean prediction.

5.4 Abbreviations

AUC Area under ROC curve

CHF Cummulative hazard function

CoxPH Cox Proportional hazard model

CV Cross validation

dmfs Distant metastatis free survival

ER Estrogen receptor

FISH Fluorescence In Situ Hybridization

fpr False positive rate

GP Gaussian Process

HER2 Human epidermal growth factor receptor

HR Hazard ratio

IHC Immunohistochemistry

NPI Nottingham prognostic index

OOB Out Of Bag sample

PgR Progesterone receptor

RF Random forest

rfs recurrence free survival

ROC Receiver operating characteristics

RSF Random survial forest

tpr True positive rate

STG St. Gallen consensus criteria

TSP Top scoring pair