BACHELORARBEIT

Auswirkung der Parameterwahl auf die Variablenselektion im R-Paket „rbsurv“

Institut für Statistik
Institut für Medizinische Informationsverarbeitung,
Biometrie und Epidemiologie
LMU München

Autor: Christian Reinhold Bihl
Betreuer: Prof. Dr. Anne-Laure Boulesteix
Nicole Schüller, MSc
Datum: 5. März 2018

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Im Rahmen dieser Bachelorarbeit geht es um die Funktion *rbsurv*, die im Statistik-Programm `R` implementiert ist. Die Funktion findet hauptsächlich in der Biologie ihre Verwendung, genauer gesagt bei der Auswertung von Microarray-Daten. Das Ziel von Auswertungen mit der Funktion *rbsurv* ist das Entdecken von Genen, welche die Überlebenszeit von Individuen beeinflussen. Der darin enthaltene Algorithmus basiert dabei hauptsächlich auf dem *Cox proportional hazards model* und verspricht laut den Autoren Cho et al. (2009) eine einfache und praktische Anwendung, die dennoch robuste Schätzungen und Ergebnisse liefert. Um eine auf die Daten angepasste Auswertung vorzunehmen, kann der Anwender verschiedene Parameter der Funktion nach seinen Bedürfnissen verändern. Um die Parameter-Einstellungen und ihre Auswirkungen aufzuzeigen, wird ein multizentrischer Datensatz von Kopf-Hals-Tumor-Patienten verwendet. Pro Patient liegen dafür ca. 1000 Expressionen von Micro-RNA's vor, die eventuell in Verbindung zu dem Tumor stehen könnten.

Das Ziel dieser Bachelorarbeit ist die Untersuchung und Bewertung der *rbsurv*-Funktion und ihrer Ergebnisse. Dabei steht vor allem die Parameterwahl und die dadurch verursachte Variablenselektion im Mittelpunkt. Im Zuge dessen wird versucht, ein optimales Modell durch eine optimale Parametereinstellung zu finden. Als Modellgütekriterium wird hierfür der Konkordanz-Index nach Harrell et al. (1982) verwendet.

Nach der Einleitung folgt in Kapitel 2 eine kurze Erklärung zur Survival Analyse und ihren Grundlagen. Hierbei wird unter anderem das *Cox proportional hazards model* genauer vorgestellt, das im R-Paket *rbsurv* verwendet wird. In Kapitel 3 geht es ausschließlich um das R-Paket *rbsurv*. Dabei wird der Algorithmus der *rbsurv*-Funktion und die dazugehörigen Parameter genauer betrachtet. Die Analyse folgt in Kapitel 4. Nach der Vorstellung der Daten und dem angewandten Vorgehen befinden sich hier zwei Simulationsläufe mit jeweils sechs Iterationen. Die Ergebnisse jener Simulationen im Bezug auf die Variablenselektion und die Robustheit des R-Paketes *rbsurv* sind in Kapitel 5 enthalten. Kapitel 6 schließt die Bachelorarbeit dann mit einem kurzen Fazit ab. Weitere Abbildungen und Tabellen sind zudem noch im Anhang (Kapitel 7) zu finden.

2 Survival-Analyse

Die Survival-Analyse (oder auch *Überlebenszeitanalyse/Ereigniszeitanalyse*) untersucht die (Lebens-) Zeit unterschiedlicher Abläufe bis ein vorher festgelegtes Ereignis (Event) auftritt. Verwendung findet die Survival Analyse in der medizinischen und biologischen Forschung, bei der Entwicklung verschiedener Produkte oder der Analyse von demographischen Gegebenheiten. Voraussetzung für die Anwendung der Survival Analyse ist, dass die Objekte innerhalb des Beobachtungszeitraums einem Risiko für das Eintreten eines Events ausgesetzt sind. Dieses Event kann z.B. der Tod, das Auftreten einer Krankheit oder der Funktionsverlust eines Gerätes sein (Wollschläger, 2017).

2.1 Zensierte Daten

Survival-Daten liegen oft in zensierter Form vor. Dabei wird zwischen unterschiedlichen Varianten unterschieden. Am häufigsten tritt die sogenannte Rechts-Zensur auf. Dabei weisen die Objekte innerhalb des Beobachtungszeitraums kein Event auf, da sie z.B. aus der Studie ausgeschieden sind oder weiterhin leben bzw. gesund sind. Dennoch kann die Information der Beobachtungen in die Modellschätzung miteinbezogen werden(Liu, 2012). Zusätzlich gibt es noch Links-zensierte Daten (Event tritt vor dem Beobachtungszeitraum auf) und Intervall-zensierte Daten (Event tritt an unbekannter Stelle innerhalb eines Zeit-Intervalls auf). "Wichtig für die Survival-Analyse ist die Annahme, dass der zur Zensierung führende Mechanismus unabhängig von Einflussgrößen auf die Überlebenszeit ist"(Wollschläger, 2017, S.351).

2.2 Allgemeine Annahmen

Geht man von einer Stichprobenpopulation von N Individuen aus, so kann entweder die Zeit bis zu einem Event oder einer Zensierung beobachtet werden. Bei einer rechtszensierten Beobachtung weiß man also lediglich, dass die Zeit bis zu einem Event größer ist als der Beobachtungszeitraum. Die Überlebensfunktion lässt sich darstellen durch:

$$S(t) = P(T > t), \qquad t > 0, \tag{1}$$

mit
$S(t)$ = Überlebensfunktion
T = pos. Zufallsvariable für den Zeitpunkt eines Events
t = Zeit (Nikulin und Wu, 2016).

Sie ist die Wahrscheinlichkeit dafür, dass ein beliebiges Individuum aus den Daten den Zeitpunkt t überlebt (Nikulin und Wu, 2016). Folglich ergibt sich die kumulative Verteilungsfunktion der Lebenszeit T mit

$$F(t) = P(T \leq t) = 1 - S(t) \tag{2}$$

(Nikulin und Wu, 2016).

Die Hazard-Funktion drückt letztlich die unmittelbare Ereignisrate einer Beobachtung zum Zeitpunkt t aus:

$$\begin{aligned} \lambda(t) &= \lim_{\Delta_t \to 0^+} \frac{P(t \leq T < t + \Delta_t | T \geq t)}{\Delta_t} \\ &= \lim_{\Delta_t \to 0^+} \frac{P(t \leq T < t + \Delta_t)/\Delta_t}{P(T > t)} \\ &= \frac{f(t)}{S(t)}, \qquad t \geq 0 \end{aligned} \tag{3}$$

mit
$f(t)$ = Dichtefunktion der Überlebenszeit T (Wollschläger, 2017).

2.3 Cox proportional hazards model

Die Cox-Regression ist ein nach Sir David Roxbee Cox benanntes Analyseverfahren für Ereigniszeitdaten. Es handelt sich dabei um ein semi-parametrisches Regressionsmodell, dessen Modellgleichung sich folgendermaßen ergibt:

$$\lambda(t; z) = exp(z^T \beta)\lambda_0(t), \tag{4}$$

mit
t = Zeit
z = Vektor der Einflussvariablen
β = Vektor der Parameterschätzer
$\lambda_0(t)$ = baseline hazard (allgemeine Ausfallrate) (Cox, 1972).

Das Modell macht dabei keine Annahmen über die Form des *baseline hazard*. Falls das Modell keine Einflüsse enthält ($z = 0$), so bleibt lediglich der *baseline hazard* $\lambda_0(t)$ übrig. Dieser spiegelt das Grundrisiko der Beobachtungen wider. Dabei wird auch für die beobachtete Überlebenszeit T keine bestimmte Verteilung angenommen. Stattdessen nimmt man an, dass die Effekte verschiedener Variablen auf das Überleben über die Zeit konstant sind (Ziegler et al., 2004). Durch diese Annahme ergeben sich sowohl Vor- als auch Nachteile. Durch die konstanten Effekte über die Zeit hinweg lässt sich der Hazard Ratio, also der Quotient zweier Hazard-Funktionen, eindeutig definieren und interpretieren. Allerdings ist die Annahme dieser Proportionalität der Hazard-Funktionen nicht immer korrekt. So kann es in der Realität durchaus eine von der Zeit abhängige Variable geben, wie z.B. das Gewicht oder das Alter einer Person (Ziegler et al., 2004). Nimmt man beispielsweise an, dass das Cox Modell lediglich eine Einflussvariable besitzt und somit die folgende Form aufweist:

$$\lambda(t; z) = exp(z_1 \beta_1)\lambda_0(t), \tag{5}$$

mit
t = Zeit
z_1 = Einflussvariable
β_1 = Parameterschätzer
$\lambda_0(t)$ = baseline hazard (allgemeine Ausfallrate).

Dann ist der erwartete Hazard Ratio (Risikoquotient) bei zwei unterschiedlichen Beobachtungen A und B

$$\frac{exp(z_A\beta_1)\lambda_0(t)}{exp(z_B\beta_1)\lambda_0(t)} = exp((z_A - z_B)\beta_1), \tag{6}$$

und damit unabhängig von der Zeit t und dem *baseline hazard*.

2.3.1 Partielle Likelihood-Funktion

Für die Schätzung und Interpretation eines Modells sind in erster Linie die β-Parameterschätzer von Interesse. Dadurch, dass der *baseline hazard* eine willkürliche Störgröße darstellt, kann man allerdings keine normale Maximum-Likelihood-Methode (ML-Methode) anwenden. In diesem Zusammenhang schlägt Cox (1972) eine partielle Likelihood-Schätzung vor. Er argumentiert damit, dass die Zeitintervalle zwischen den Events keine wichtigen Informationen liefern, da die Störgröße $\lambda_0(t)$ dort vermutlich identisch mit Null ist. Bei Survival-Daten ohne Bindungen (siehe Abschnitt 2.3.2) reicht es also aus, wenn nur die Zeitpunkte betrachtet werden, in denen ein Event stattfindet. So ist für ein bestimmtes Event zum Zeitpunkt $t_{(i)}$, bedingt auf die Risikomenge $R(t_{(i)})$, die Wahrscheinlichkeit, dass das Event auch beim beobachteten Individuum auftritt, folgende:

$$\frac{exp\{z_{(i)}\beta\}}{\sum_{l \in R(t_{(i)})} exp\{z_{(l)}\beta\}} \tag{7}$$

(Cox, 1972).

Aus dem Produkt der einzelnen Beobachtungen ergibt sich damit die partielle Likelihood:

$$L(\beta) = \prod_{i=1}^{N} \frac{exp\{z_{(i)}\beta\}}{\sum_{l \in R(t_{(i)})} exp\{z_{(l)}\beta\}} \tag{8}$$

bzw. die partielle log-Likelihood:

$$\ell(\beta) = \sum_{i=1}^{N} z_{(i)}\beta - \sum_{i=1}^{N} log\left[\sum_{l \in R(t_{(i)})} exp\{z_{(l)}\beta\}\right] \tag{9}$$

(Cox, 1972).

Auch wenn es sich hierbei nicht um eine gewöhnliche Likelihood-Schätzung handelt, so kann sie doch in den meisten Fällen als eine solche behandelt werden. So besitzt die ML-Schätzung nach Cox in großen Datenmengen die selben asymptotischen Eigenschaften wie eine normale ML-Schätzung (Kalbfleisch und Prentice, 2002).

2.3.2 Bindungen

In den meisten Survival-Daten lassen sich allerdings Bindungen finden. Diese Bindungen enstehen, wenn Beobachtungen die exakt selbe Überlebenszeit besitzen. Würde man die Zeit auf einer perfekten stetigen Skala messen, würde dieser Fall nie eintreten. Allerdings wird in der Realität der Einfachheit halber meist eine diskrete Zeit-Skala verwendet (Borucka, 2014). Auch durch Zensierungen kann es zu vielen gleichen Überlebenszeiten kommen. Da bei der partiellen Likelihood-Schätzung nach Cox jedoch die Reihenfolge der Events von Bedeutung ist, kann dies bei Bindungen zu Problemen führen. Die partielle Likelihood muss dann dementsprechend angepasst werden. Der natürlichste Weg ist, laut Kalbfleisch und Prentice (2002), die durchschnittliche Likelihood zu berechnen, die sich aus allen möglichen Kombinationen aus der Reihenfolge der Events ergibt. Bei einer großen Anzahl an Bindungen führt diese Methode allerdings zu einem hohen Rechen-, bzw. Zeitaufwand. Aufgrund dessen gibt es mehrere approximative Alternativen für die partielle Likelihood, welche bei Bindungen einen geringeren Zeitaufwand versprechen. Die am häufigsten verwendeten Methoden sind neben der exakten Methode von Kalbfleisch und Prentice (2002) die Methoden nach Breslow (1974) und Efron (1977). Die resultierenden Schätzer durch die Methoden von Breslow und Efron können je nach Stichprobengröße und Anzahl an Bindungen eine Verzerrung aufweisen. Für Datensätze mit geringer Anzahl an Bindungen erreichen alle drei Schätzer ähnliche Resultate (Kalbfleisch und Prentice, 2002). Auch wenn die Methoden nach Breslow und Efron eine Verzerrung der Schätzer bewirken können, so werden sie in der Praxis dennoch häufig angewendet, da sie im Vergleich zur exakten Methode von Kalbfleisch und Prentice (2002) einen deutlich geringeren Rechenaufwand mit sich bringen.

2.4 C-Index

Nachdem ein geeignetes Modell angepasst wurde, stellt sich die Frage, wie gut das Modell die Wirklichkeit abbildet. Durch falsche Annahmen, fehlende Daten oder durch nicht miteinbezogene Störvariablen kann es erhebliche Verzerrungen im Modell geben. Der Konkordanz-Index C (Harrell et al., 1982) ist dabei ein oft verwendetes Validierungswerkzeug für Überlebenszeitmodelle. Für unzensierte Daten stellt der Konkordanz-Index C (c-Index) die relative Häufigkeit von konkordanten Paaren unter allen möglichen Paaren dar. Dabei wird ein Paar als konkordant bezeichnet, wenn das Individuum mit der geringeren Überlebenszeit auch das höhere Risiko für ein Event besitzt (Gerds et al., 2013). Ist die vom Modell prognostizierte Überlebenszeit für zwei Individuen identisch, so werden sie nur zur Hälfte mitberechnet. Besitzen zwei Individuen dagegen die selbe Überlebenszeit, so gelten sie als unbrauchbar und können nicht verwendet werden (Harrell et al., 1996).
Der c-Index kann Werte zwischen 0 und 1 annehmen. Dabei entspricht ein Wert von 1 einem perfekten Modell bzw. einer perfekten Vorhersage. Nimmt der c-Index einen Wert von 0.5 an, so ist das Modell nicht besser als eine willkürliche Zufallsentscheidung (Harrell et al., 1996).
Um die Güte eines Modells durch den c-Index zu berechnen, wird zusätzlich zu den Daten, die für die Modellberechnung verwendet wurden, ein weiterer Datensatz benötigt. Dieser sollte unabhängig von den anderen Daten sein, um somit die Prognosefähigkeit des Modelles testen zu können.

3 R-Paket *rbsurv*

Dieses Kapitel basiert hauptsächlich auf dem im Januar 2009 von HyungJun Cho et. al. vorgestellten Artikel im *Journal of Statistical Software* über ihr neues R-Paket *rbsurv* (Cho et al., 2009). Das R-Paket dient dazu multiple Survival-Modelle aus Microarray-Daten zu bilden. Hierbei wird das Software-Programm R verwendet, das in der Statistik sehr verbreitet ist. Das Paket bzw. der darin enthaltene Algorithmus basiert auf der Cox-Regressionsanalyse. Mit Hilfe des Software-Paketes lassen sich Gene finden, die einen Bezug zur Überlebenszeit eines Individuums besitzen.

3.1 Hintergrund und verwendete Methoden

Bei der Analyse von Microarray-Daten werden häufig Daten mit hoher Dimension und geringer Stichprobengröße verwendet (Engler und Li, 2009). Um aus dieser hohen Anzahl an Variablen, die mit dem größten Einfluss herauszufiltern, wurden in der Vergangenheit bereits viele verschiedene Verfahren angewendet (vgl. dazu Rosenwald et al., 2002; Shannon et al., 2002; Gui und Li, 2005). Trotz dieser bereits existierenden Algorithmen entschieden sich die Autoren für eine neue Variante, die im Software-Programm R eingebunden ist. Der verwendete Algorithmus kann dabei aktiv vom Benutzer verändert bzw. angepasst werden und verspricht eine robuste Schätzung aufgrund der verwendeten Kreuzvalidierungstechnik.

3.1.1 Kreuzvalidierung

Die Kreuzvalidierung ist ein statistisches Verfahren, um die Güte eines Modells zu überprüfen. Bei der Kreuzvalidierung wird meist der Datensatz in einen Trainings- und einen Testdatensatz eingeteilt. Dabei ist es oft besser, wenn der Trainingsdatensatz mehr als 50 Prozent der Daten enthält. Verbreitet ist, dass der Trainingsdatensatz zwei Drittel der Datenmenge enthält (Witten et al., 2016). Der Trainingsdatensatz bildet letztlich die Grundlage für die Modell- bzw. Parameterschätzung. Mit Hilfe des Testdatensatzes wird dann die Fehlerrate des Modells berechnet. Verschiedene Modifikationen des Verfahrens wie z.B. die stratifizierte Kreuzvalidierung bauen auf dem selben Grundprinzip auf.

3.1.2 Akaikes Informationskriterium

Ein Gütekriterium, das auch im R-Paket *rbsurv* verwendet wird, ist das Akaike Informationskriterium (*AIC*). Das AIC wurde entwickelt, um einen möglichst guten Kompromiss zwischen einer guten Datenanpassung und einer zu großen Modellkomplexität zu finden (Fahrmeir et al., 2007). So wird durch die Hinzunahme von (unnötig) vielen Variablen das Modell überangepasst (*engl.: overfitting*) und somit eventuell die Prognosefähigkeit verschlechtert (Fahrmeir et al., 2007). Das AIC lässt sich durch die Formel

$$\mathrm{AIC} = -2l(\hat{\theta}) + 2p \tag{10}$$

darstellen. Es gilt:
$\hat{\theta}$ = p-dimensionaler Parametervektor
$l(\hat{\theta})$ = log-Likelihood der geschätzten Parameter
p = Anzahl der geschätzten Parameter (Fahrmeir et al., 2007).

Letztlich wird dasjenige Modell bevorzugt, das den geringsten AIC aufweist. Der Term $2p$ bestraft somit die Modelle proportional zur Anzahl der enthaltenen Parameter p.

3.2 Algorithmus

Um einen effektiven und übersichtlichen Algorithmus für die Analyse von Microarray-Daten zu ermöglichen, haben sich die Autoren des Paketes *rbsurv* verschiedene Schritte überlegt. Dabei sollten die Einflussvariablen bereits normalisiert und entsprechend transformiert worden sein.

1. **Beschränkung der Anzahl an Genen**

   Die meist große Anzahl von Genen in Microarray-Daten führt dementsprechend zu langen Berechnungszeiten. Um diese Berechnungszeiten möglichst kurz zu halten, enthält das Software-Programm eine Art Vor-Selektion der wichtigsten Gene. Dafür werden univariate Überlebenszeitmodelle genutzt, die dann die Anzahl der Gene reduzieren ohne wichtige Gene bzw. Informationen zu verlieren. Das bedeutet, es wird für jedes Gen ein univariates Überlebenszeitmodell gebildet und anschließend werden die Gene ausgewählt, die in den Modellen den kleinsten p-Wert aufweisen. Die Auswahl der wichtigsten Gene erfolgt, wenn vom Anwender gewünscht, zu Beginn der Auswertung und wird mit dem Befehl `[max.n.genes]` übergeben. Nur diese gewünschte Anzahl an Genen wird dann auch in den Algorithmus miteinbezogen.

2. **Robuste Likelihood-Schätzung in Überlebenszeitmodellen**

   Der Datensatz wird zufällig in einen Trainings- und einen Validierungsdatensatz eingeteilt. Die Größe der zwei Teil-Datensätze wird mit dem Befehl `[n.fold = 1/p]` angegeben. Dabei enthält der Trainingsdatensatz $N(1-p)$ und der Validierungsdatensatz Np Beobachtungen. Anschließend wird für jedes Gen getrennt der Maximum-Likelihood-Schätzer $\hat{\beta}_i^0$ auf Grundlage des Trainingsdatensatzes berechnet. Um einen robusten Schätzer zu erhalten ist es von Vorteil, das Modell durch einen unabhängigen Datensatz zu evaluieren, anstatt mit den Daten, die zur Parameterschätzung verwendet wurden. Diesen unabhängigen Datensatz stellt der davor zufällig abgetrennte Validierungsdatensatz dar. Die angepasste partielle log-Likelihood dient mit der Verwendung des Validierungsdatensatzes damit als Gütemaß für die Anpassung des Modells.

3. **Robuste Genschätzungen**

   Die Prozedur in Punkt 2 wird anschließend **B**-mal wiederholt. Dies wird der Funktion mit Hilfe des R-Befehls `[n.iter = B]` mitgeteilt. Damit wird das Risiko, dass das zufällige Einteilen in Trainings- und Validierungsdatensatz eine Verzerrung erzeugt, minimiert. Aus den, für jedes einzelne Gen, erhaltenen **B** partiellen log-Likelihoods wird das Gen mit der größten mittleren partiellen log-Likelihood ausgewählt. Dieses Gen *(hier: $g_{(1)}$)* hat nach diesem Ansatz den größten Einfluss auf die Überlebenszeit.

4. **Robuste Modellselektion**

   Nachdem das ausgewählte Gen $g_{(1)}$ dem Modell übergeben wird, werden die Punkte 2 und 3 abermals durchgeführt, um das beste Modell mit zwei Genen als Einflussvariablen zu finden. Es wird somit ein Modell gesucht, das zu dem bereits ausgewählten Gen $g_{(1)}$ ein bestmögliches zweites Gen $g_{(2)}$ findet. Diese Vorwärts-Selektion wird so lange durchgeführt, bis es durch fehlende Beobachtungen nicht mehr möglich ist ein entsprechendes Modell anzupassen oder die maximale Anzahl an Genen erreicht ist. Man erhält letztlich K Modelle mit: $M_1 = g_{(1)}, M_2 = g_{(1)} + g_{(2)}, \ldots, M_K = g_{(1)} + g_{(2)} + \cdots + g_{(K)}$.
   Um das beste Modell auszuwählen, ist die log-Likelihood nicht geeignet, da sie in jedem Fall das größte Modell auswählen würde. Um ein *overfitting* zu vermeiden, wird das AIC verwendet. Durch den darin enthaltenen Strafterm für die Anzahl der Variablen wird somit ein Modell mit geringer Variablenzahl bevorzugt. Es wird das AIC für jedes der Modelle berechnet und das Modell mit dem kleinsten AIC wird anschließend ausgewählt.

5. **Multiple Modelle**

   Bei der Selektion der Gene für das optimale Modell können aufgrund des Algorithmus wichtige Gene fehlen. Angenommen zwei Gene haben einen ähnlichen Effekt auf die Überlebenszeit. Statistisch gesehen reicht es, das stärker assoziierte Gen auszuwählen. Biologisch betrachtet kann das zweite Gen allerdings ebenfalls einen wichtigen Einfluss auf die Überlebenszeit besitzen. Um dieses Szenario zu verhindern, kann man mehrere optimale Modelle berechnen. Dafür werden die Gene der ersten Modellberechnung zur Seite genommen und mit den restlichen verbliebenen Variablen ein zweites Modell berechnet. Die Anzahl an Modellberechnungen ist dem Benutzer überlassen (vorausgesetzt es sind genügend Gene vorhanden). Dementsprechend ist statistisch gesehen das erste Modell das Beste, aber nicht unbedingt aus der biologischen Sicht. Die Anzahl der berechneten Modelle wird im R-Paket durch `[n.seq = C]` angegeben.

3.3 Risikofaktoren

Die Überlebenszeit hängt nicht zwingend nur mit dem untersuchten Genmaterial zusammen. So kann es weitere Risikofaktoren wie z.B. das Alter oder der Krankheitsstatus geben, für die das Modell adjustiert werden sollte. Dadurch könnte eventuell ein Gen ins Modell aufgenommen werden, welches eigentlich die Risikofaktoren beeinflusst aber nicht direkt die Überlebenszeit. Um die dadurch entstehenden Verzerrungen zu verhindern, kann man dem Algorithmus für die likelihood-basierte Modellschätzung zusätzliche Risikofaktoren übergeben. Diese Faktoren $Z_1, Z_2, \ldots, Z_p$ werden in alle Modell-Anpassungen im Algorithmus miteinbezogen [*rbsurv:* ***z***]. Mit dem Befehl `[alpha]` kann ein Signifikanzlevel für die Risikofaktoren angegeben werden (z.B. `[alpha = 0.05]`). Somit werden lediglich signifikante Risikofaktoren verwendet. Im Zuge dieser Auswertung sind allerdings keine weiteren Risikofaktoren gegeben und dementsprechend wurde dieser Parameter auch nicht weiter betrachtet.

3.4 Sonstige Argumente in *rbsurv*

Weitere Argumente der Funktion *rbsurv* sind (vgl.: Tabelle 1): `[method]`, `[gene.ID]` und `[seed]`. `[method]` gibt dabei an, welche Berechnungsmethode für Bindungen bei der Cox-Regression verwendet werden. Hierbei wird unterschieden zwischen *"breslow", "efron"* und *"exact"*. Mit dem Befehl `[gene.ID]` kann der Benutzer den Genen einen Namen oder eine Identifikationsnummer zuordnen. Falls dies nicht explizit angegeben ist, wird die Reihennummer verwendet. Mit `[seed]` wird der Zufallsgenerator auf einen festen Startwert gesetzt. Dies bedeutet, dass die zufällige Einteilung in Trainings- und Validierungsdatensatz im Algorithmus bei gleichem `[seed]` auch immer gleich ist. Dies ist nützlich, um die Ergebnisse reproduzieren bzw. die Variabilität der Ergebnisse betrachten zu können.

Argument	Beschreibung
time	Vektor mit den Überlebenszeiten
status	Vektor mit Status (0 = zensiert, 1 = Event/Krankheit wieder aufgetreten)
x	Matrix mit den Einflusswerten (Gene in Reihen, Beobachtungen in Spalten)
z	Matrix für die zusätzlichen Risikofaktoren
alpha	Signifikanzlevel für die Risikofaktoren
gene.ID	Vektor mit den Gen-IDs, ansonsten werden Reihennummern verwendet
method	character string um die Methode für Bindungen festzulegen
n.iter	Anzahl an Iterationen bei der Genselektion
n.fold	Anzahl an Partitionen der Beobachtungen
n.seq	Anzahl an multiplen Modellen
seed	seed für Einteilung der Beobachtungen
max.n.genes	maximale Anzahl an betrachteten Genen

Tabelle 1: Übersicht der Argumente im R-Paket *rbsurv*

4 Analyse

Im Hauptteil dieser Bachelorarbeit geht es um den verwendeten Datensatz und die damit durchgeführten Simulationen. Um die Variabilität der Variablenselektion aufgrund der Parameterwahl aufzuzeigen, wurden die Simulationen nach einem festen Vorgehen durchgeführt.

4.1 Erklärung der Daten

Der für diese Auswertung verwendete Datensatz stammt aus zwei unabhängigen Kohorten von strahlentherapeutisch behandelten Kopf-Hals-Tumor-Patienten. Insgesamt enthält der Datensatz 162 Beobachtungen von Patienten aus Deutschland. Dabei stammen 85 Personen aus einer multizentrischen Kohortenstudie vom Deutschen Konsortium für translationale Krebsforschung (DKTK) und 77 Personen von einer monozentrischen Kohorte aus der klinischen Kooperationsgruppe (KKG) der LMU München und der Klinik für Strahlentherapie und Radioonkologie. Die genaue Aufteilung der Kohortenstudien sowie die im weiteren Verlauf verwendeten Abkürzungen finden sich in Tabelle 2. Aus dem Genmaterial dieser 162 Patienten wurden globale micro-RNA (miRNA) Expressionsanalysen durchgeführt. MiRNA's sind kleine, hoch konservierte, nicht-kodierende RNA-Moleküle, die an der Regulation der Genexpression beteiligt sind (MacFarlane und R Murphy, 2010). Da Änderungen an den miRNA's Auswirkungen auf einen menschlichen Tumor haben können (Calin und Croce, 2006), enthält dieser Datensatz 1031 verschiedene miRNA's mit den dazugehörigen Expressionen. Hierfür wurden miRNA-Proben mit dem Fluoreszenzfarbstoff Cy3 markiert. Diese hybridisieren mit den jeweils komplementären miRNA-Sequenzen auf einem Array und anhand der Intensität des Fluoreszenzsignals wird dann die Expression der miRNA's gemessen (Lohaus et al., 2014). Damit könnte man im Idealfall Rückschlüsse ziehen, welche miRNA's das Wiederauftreten des hier untersuchten Kopf-Hals-Tumors begünstigen. Um übersichtliche Abbildungen zu ermöglichen werden im weiteren Verlauf die Bezeichnungen der miRNA's durch Identifizierungsnummern abgekürzt. Im Anhang befindet sich dazu in Tabelle 7 die Gegenüberstellung der Namen und ihrer ID-Nummern. Zusätzlich zu den Expressionen der 1031 miRNA's liegen noch die Informationen über den Status der Person und ihre beobachtete Überlebenszeit vor. Der Status der Patienten ist dabei binär codiert und bedeutet, dass bei dem Patient mit [`Status = 0`] bis zum Ende der Studie der Tumor nicht wieder aufgetreten ist. Patienten mit [`Status = 1`] dagegen sind dementsprechend Personen, bei welchen der Tumor noch innerhalb des Studienzeitraumes wieder aufgetreten ist. Die Überlebenszeit ist in diesem Fall nicht wörtlich zu nehmen. Stattdessen entspricht sie entweder der Zeit bis zum Wiederauftreten des Tumors, bis zum Studienende oder bis zum Ausscheiden des Patienten aus der Studie. Die Überlebenszeit ist dabei in Tagen angegeben und reicht von minimal 56 Tagen bis zu maximal 3002 Tagen.

4.2 Deskription

Aufgrund der Größe des Datensatzes ist es im Rahmen dieser Arbeit nicht möglich, jede einzelne Variable genauer vorzustellen. Es werden lediglich die Wichtigsten betrachtet. Dazu gehören sowohl die beobachtete Überlebenszeit, welchen Status sie aufweist und aus welchem Institut die Person stammt. Zusätzlich werden noch einige repräsentativen miRNA's näher betrachtet.

Kohorte	Abkürzung	Institutsort
DKTK	BER	Berlin
DKTK	DD	Dresden
DKTK	EU	Essen
DKTK	FB	Freiburg
DKTK	FFM	Frankfurt am Main
DKTK	HD	Heidelberg
DKTK	TUE	Tübingen
DKTK	TUM	Technische Universität München, Klinikum rechts der Isar
KKG	KKG	Klinische Kooperationsgruppe
DKTK	DKTK	Deutsches Konsortium für Translationale Krebsforschung

Tabelle 2: Erklärung der Abkürzungen für die Institute.

Der Datensatz enthält 162 Beobachtungen, die aus verschiedenen Instituten in Deutschland stammen. Abbildung 1 zeigt dabei die Verteilung auf die insgesamt neun Institute. Dabei ist eine sehr ungleiche Verteilung zu beobachten. So stammen allein 77 Patienten aus der klinischen Kooperationsgruppe LMU/Helmholtz Zentrum München, während aus Berlin und Heidelberg lediglich je zwei Patienten kommen. Da sich die Stichprobe aus sehr vielen verschiedenen Quellen zusammensetzt, kann es dadurch auch zu Verzerrungen kommen. So können in den Krankenhäusern z.B. unterschiedliche Standards in der Krebsbehandlung vorliegen. Auch die Qualität der behandelnden Ärzte ist nicht überall gleich und kann sich somit auf das Wiederauftreten des Krebs und damit auf die Ergebnisse der Analyse mit auswirken.

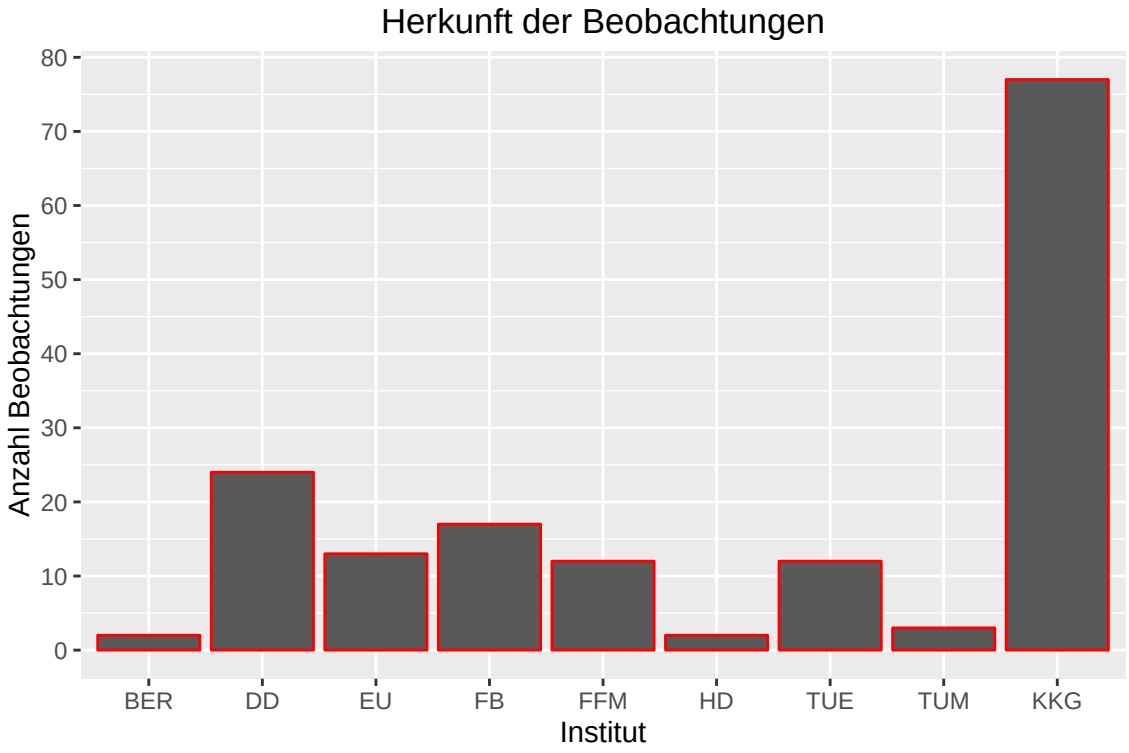

Abbildung 1: Die Anzahl der Krebspatienten, welche in der Studie beobachtet wurden, pro Institut.

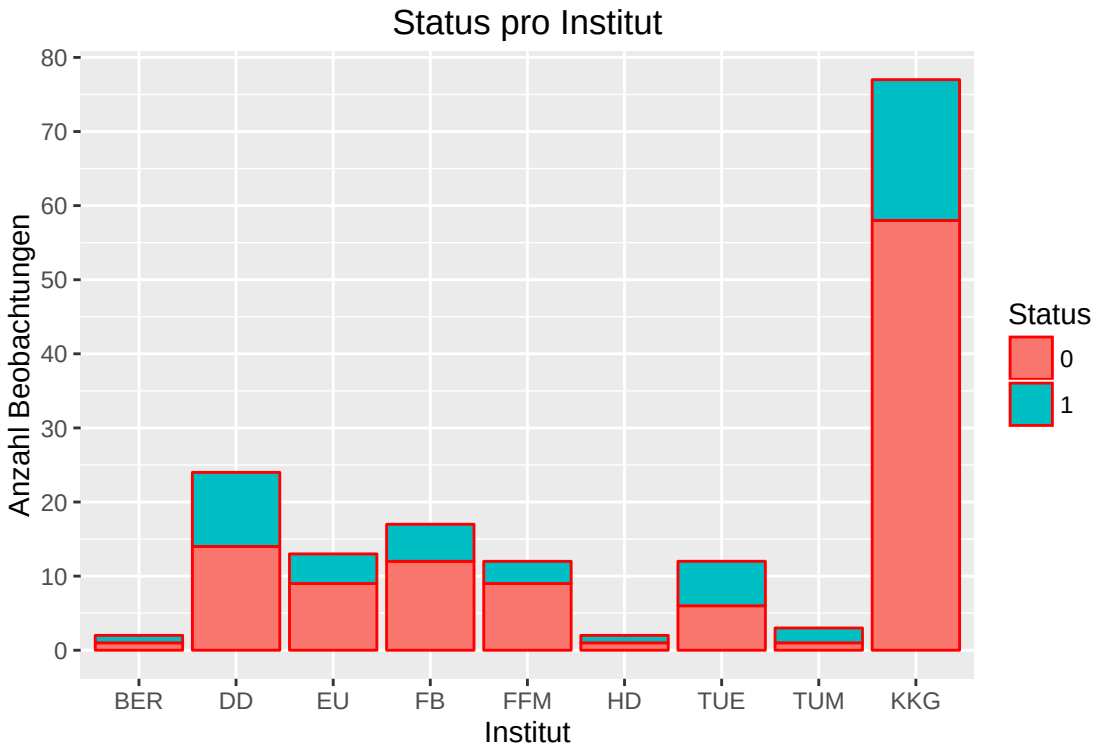

Abbildung 2: Die Anzahl der Krebspatienten, welche in der Studie beobachtet wurden, bei denen der Krebs zurückkam (Status = 1) und die zensiert wurden (Status = 0).

Abbildung 2 zeigt zusätzlich zur Herkunft der Patienten das Verhältnis zwischen Personen mit dem Wiederauftreten von Krebs (gekennzeichnet mit [`Status = 1`]) im Gegensatz zu zensierten Personen (gekennzeichnet mit [`Status = 0`]). Hierbei ist zu sehen, dass aus jedem Krankenhaus mindestens eine Person mit Status 1 und eine Person mit Status 0 kommt. Das Verhältnis ist dabei aber nicht immer identisch. Bis auf das Institut der TUM sind allerdings immer mindestens 50% der Beobachtungen aus den Krankenhäusern zensiert ([`Status = 0`]).
Auch die Verteilung der Überlebenszeit (*hier:* Zeit bis zum Event bzw. bis zur Zensur) ist von Bedeutung. In Abbildung 3 sind die Zeiten in Form von Boxplots dargestellt. Bis auf das Krankenhaus HD weisen alle Gruppen einen relativ ähnlichen Mittelwert auf. Die Länge der Boxen, welche die mittleren 50% der Daten enthalten, ist dagegen sehr abhängig von der Gruppengröße. Besonders die Krankenhäuser BER, HD und TUM haben dementsprechend eine geringere Streuung der Überlebenszeiten. Allerdings lässt sich allein mit dieser Abbildung noch kein Zusammenhang zwischen der Überlebenszeit und dem Status der Person feststellen.
Abbildung 4 bildet dagegen den Status der Personen in Relation zur Überlebenszeit ab. Die horizontalen Linien zeigen die jeweiligen Mittelwerte. Dabei ist der große Unterschied zwischen den zwei Mittelwerten deutlich wahrnehmbar. So liegt der Mittelwert für die Beobachtungen mit Status 0 bei circa 1535 Tagen, während er für Beobachtungen mit Status 1 nur bei ca. 432 Tagen liegt. Worin dieser Unterschied begründet liegt, ist allerdings nicht ersichtlich. Ein möglicher Grund dafür wäre, dass das Risiko für das Wiederkehren des Kehlkopfkrebses besonders in der Anfangszeit erhöht ist.

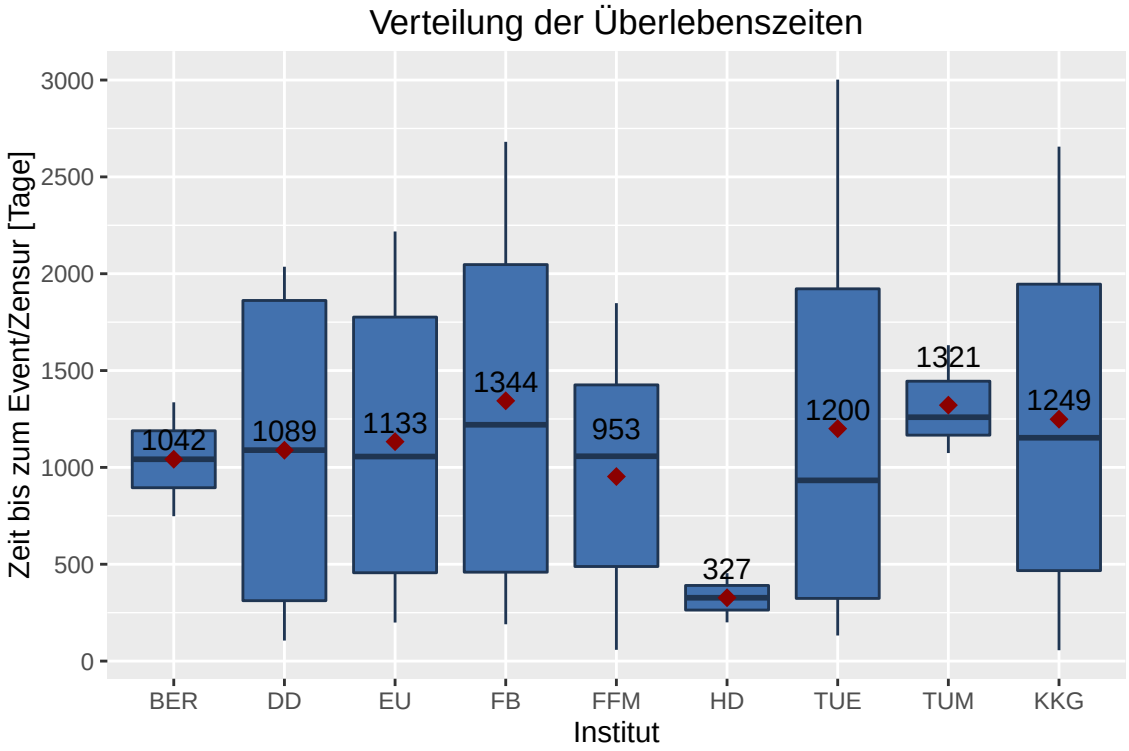

Abbildung 3: Die Verteilung der Überlebenszeiten betrachtet auf die Krankenhäuser der Krebspatienten, welche in der Studie beobachtet wurden. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten.

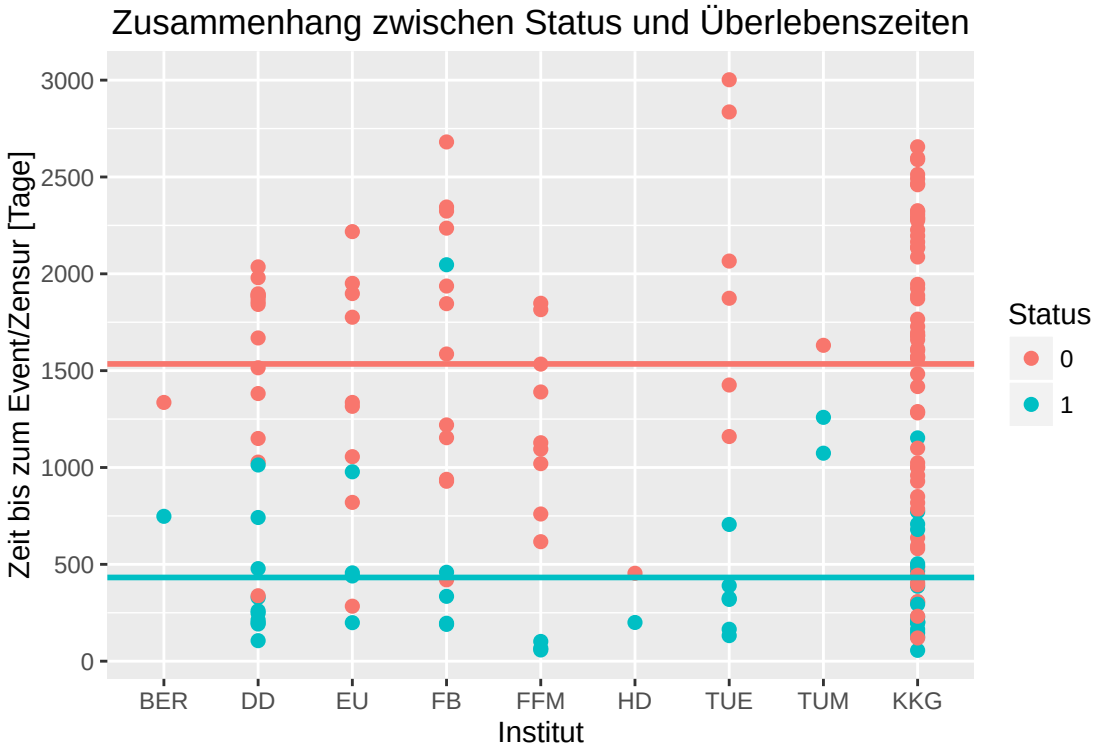

Abbildung 4: Die Verteilung der Überlebenszeiten abhängig vom Status und der Herkunft der Beobachtungen. Status = 0 steht für zensierte Beobachtungen und Status = 1 für Beobachtungen bei denen der Kopf-Hals-Tumor wiederaufgetreten ist. Die horizontalen Linien stellen die jeweiligen Mittelwerte der Beobachtungen dar.

Wie bereits erwähnt ist es nicht möglich alle 1031 Einflussvariablen näher zu betrachten. Abbildung 5 ermöglicht allerdings einen kleinen Überblick über vier ausgewählte Variablen. Die dort gezeigten Variablen sind auch diejenigen, die den größten bzw. kleinsten Wert aufweisen mit ca. 6 und -5.5. Alle Einflussvariablen wurden zudem standardisiert und besitzen dadurch einen Mittelwert von 0 und eine Varianz von 1.

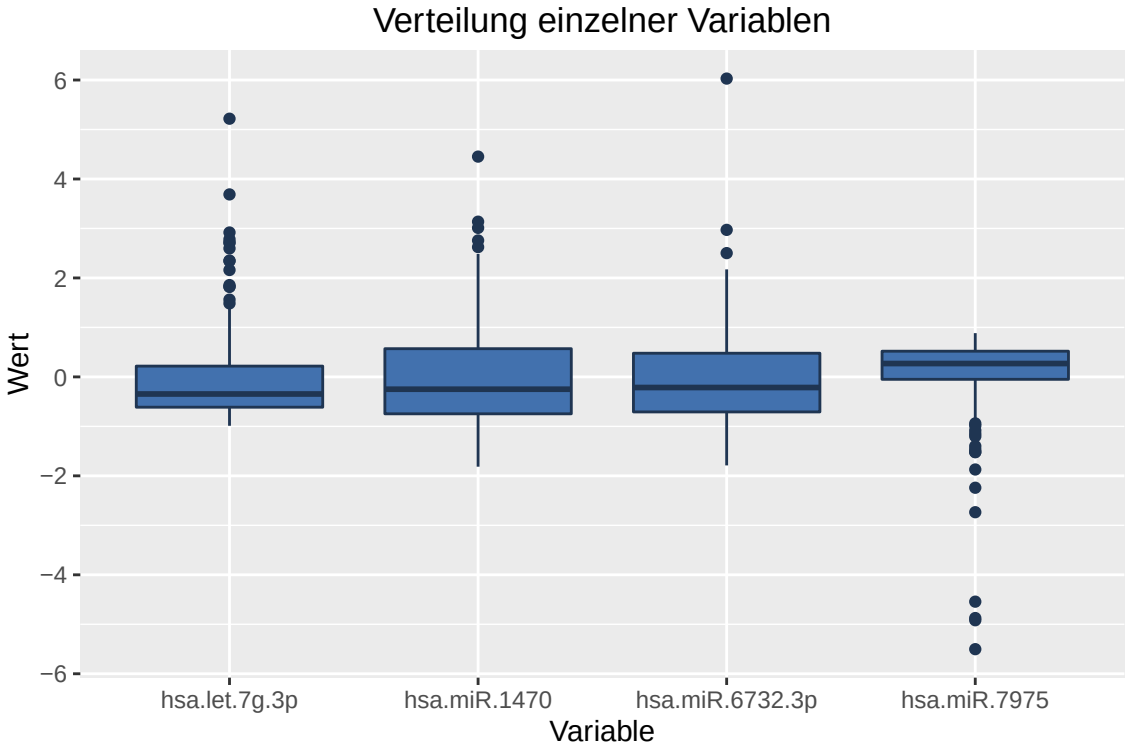

Abbildung 5: Die Verteilung einiger ausgewählter Einflussvariablen, die bereits in eine standardisierte Form gebracht wurden.

4.3 Vorgehen

Um die Auswirkungen der Parameterwahl auf die Variablenselektion zu untersuchen, wird zunächst nach der optimalen Einstellung der Parameter geschaut. Mit dieser optimalen Einstellung kann das bestmögliche Modell berechnet werden, welches dann auch die wichtigsten Einflussvariablen enthalten sollte. Im Rahmen dieser Arbeit bedeutet bestmöglich, dass das Modell auf den unabhängigen Validierungsdaten den höchsten c-Index aufweist. Der Algorithmus der *rbsurv*-Funktion dient also dazu, die entsprechend wichtigen Variablen herauszufiltern. Mit diesen Variablen wird anschließend ein Cox-Modell auf Grundlage des Trainingsdatensatzes berechnet. Dieses wird dann mit dem Validierungsdatensatz und dem darauf berechneten c-Index bewertet. Im Zuge dieses Prozesses werden dann, durch unterschiedliche Werte für die einzelnen Parameter, Vergleiche der Modelle und ihren dazugehörigen Variablen möglich. Um mit dem vorhandenen Datensatz ein geeignetes Modell mit den wichtigsten Variablen zu erhalten, wird folgendermaßen vorgegangen:

Schritt 1: Die freien Parameter der *rbsurv*-Funktion werden in einer geeigneten Reihenfolge geordnet, in welcher sie später dann entsprechend hintereinander optimiert werden.

Schritt 2: Es wird ein Modell mit den Default-Werten der Funktion berechnet. Lediglich der erste freie Parameter wird verändert. Dabei wird das Modell mit mehreren verschiedenen Werten für den Parameter berechnet und anschließend wird der Parameter auf den besten Wert festgesetzt. Die Kriterien dafür sind hauptsächlich der c-Index und die Berechnungsdauer.

Schritt 3: Schritt 2 wird für jeden freien Parameter wiederholt. Die zuvor in Schritt 2 gewonnenen optimierten Werte werden als neue Default-Werte verwendet.

Schritt 4: Die Schritte 2 und 3 werden mit den davor gewonnenen Werten als neue Default-Werte nochmals wiederholt.

Diese hier aufgeführten Schritte dienen als Übersicht für das weitere Vorgehen. Die einzelnen Schritte und die dadurch gewonnenen Informationen werden im Folgenden ausführlicher betrachtet.

4.3.1 Schritt 1

In Tabelle 1 kann eine Übersicht aller vorhandenen Parameter gefunden werden. Allerdings werden diese nicht alle in der Modellfindung benötigt. So ist der Befehl [`gene.ID`] lediglich dazu gedacht eine bessere Übersicht über die Daten zu erhalten. Für die Modellanpassung selbst spielt er aber keine Rolle. Da es in dem vorhandenen Datensatz keine zusätzlichen Risikofaktoren gibt, werden auch die Befehle [`z`] und [`alpha`] nicht benötigt. Zusätzlich werden die Befehle [`time`],[`status`] und [`x`] nur gemeinsam verändert, da sie ja jeweils zueinander gehören und nicht getrennt verändert werden können. Sie werden deshalb im weiteren Verlauf als Parameter [`Datensatz`] zusammengefasst. Die zur Modellfindung übriggebliebenen Parameter zusammen mit ihren Default-Einstellungen sind in Tabelle 3 zu finden. Die Default-Einstellung für den Datensatz wurde auf den Teildatensatz DKTK festgelegt. Der restliche Teil der Daten (KKG) wird somit anschließend zur Validierung mit Hilfe des c-Indexes verwendet.

Argument	Beschreibung	Default-Einstellung
Datensatz	Datensatz zur Modellberechnung	*hier:* DKTK
method	die Methode für Bindungen	Methode nach Efron
n.iter	Anzahl an Iterationen bei der Genselektion	10
n.fold	Anzahl an Partitionen der Beobachtungen	3
n.seq	Anzahl an multiplen Modellen	1
seed	seed für Einteilung der Beobachtungen	1234
max.n.genes	maximale Anzahl an betrachteten Genen	*hier:* Anzahl Beobachtungen

Tabelle 3: Argumente im R-Paket *rbsurv*, die in den Simulationen verändert werden

4.3.2 Schritt 2

Dieser Schritt dient dazu den besten Wert eines Parameters herauszufinden. Hierfür wird der Parameter auf mehrere unterschiedliche Werte festgelegt und für jeden dieser Werte werden 40 verschiedene Berechnungen getätigt. Diese 40 Berechnungen ergeben sich aus der zuvor festgelegten Anzahl von 40 verschiedenen `seed‘s` pro Ausprägung. Für die Auswahl der möglichen Werte für die Parameter spielten sowohl die Default-Einstellung der *rbsurv*-Funktion, die mögliche resultierende Berechnungszeit und die gegebenen Grenzen des *rbsurv*-Algorithmus eine Rolle. Die daraus gewonnenen Berechnungen sind dann letztlich die Grundlage für die Entscheidung, auf welchen Wert bzw. welche Ausprägung der Parameter gesetzt wird. Hierfür ist primär der c-Index entscheidend. Dieser ist ein häufig verwendetes Werkzeug zur Validierung eines Überlebenszeitmodelles. Es wird dabei empfohlen, wie bereits in Abschnitt 2.4 erläutert, dass der Datensatz für die Modellfindung nicht derselbe wie für die Validierung ist. Aufgrund dessen wird lediglich ein Teil-Datensatz dem Algorithmus der *rbsurv*-Funktion übermittelt, während der restliche Teil des Datensatzes zur Validierung mit dem c-Index verwendet wird. Als Default-Einstellung wird der Algorithmus der *rbsurv*-Funktion auf dem Datensatz DKTK ausgeführt, für die Validierung dann der Datensatz KKG. Somit wird die Prognosegüte des Modells auf neue Daten getestet. Allerdings stellt der c-Index nicht das einzige Kriterium für die Modellfindung dar. Auch die Berechnungsdauer ist von Bedeutung. Je nach Datensatzumfang nehmen die Berechnungen einen erheblichen Zeitumfang ein. Deshalb ist es sowohl im Rahmen dieser Arbeit, wie auch vermutlich in vielerlei anderweitiger Verwendung durchaus ein wichtiges Kriterium, um in annehmbaren Zeiten ein ausreichend gutes Modell zu finden. Bei einem gegensätzlichen Verlauf der Berechnungszeit und der Modellgüte muss somit von Fall zu Fall entschieden werden. Die in folgenden Abbildungen angegebene Berechnungszeit dient in erster Linie dazu, die verschiedenen Berechnungszeiten in ein Verhältnis zueinander zu setzen. Die Zeiten selbst können von Computer zu Computer variieren und wurden deshalb zur Vergleichbarkeit alle am selben Gerät berechnet.

4.3.3 Schritt 3

Da es nicht ausreicht, lediglich den ersten Parameter gegebenenfalls anzupassen, wird Schritt 2 für alle weiteren Parameter in einer bestimmten Reihenfolge wiederholt. Diese Reihenfolge wurde in Schritt 1 festgelegt. Dabei bauen die einzelnen Iterationen aufeinander auf. Das bedeutet, dass, falls z.B. beim ersten Parameter der Default-Wert verändert wird, diese Veränderung auch für die folgenden Parametern mit einbezogen wird. Die Iterationen sind demnach abhängig voneinander.

4.3.4 Schritt 4

Um eine robuste Schätzung zu erlangen, werden die Schritte 2 und 3 nochmals wiederholt. Die einzige Veränderung zum ersten Durchgang besteht darin, dass jetzt bereits neue optimale Werte für die Parameter festgelegt wurden und diese im zweiten Durchgang als neue Default-Einstellungen übernommen werden. Somit kann ein Parameter im zweiten Durchgang auf einen anderen optimalen Wert gesetzt werden als noch im ersten Durchgang, da die Kombination mit den anderen Parametern verändert wurde.

4.4 Simulationen

Um aus den vorhandenen Parametern eine geeignete Reihenfolge zu finden, wurden folgende Überlegungen angestellt.

- Der Parameter [`seed`] kann zwar unterschiedliche Modelle durch unterschiedliche Werte verursachen, allerdings bedeutet dies lediglich das Eingreifen in einen Zufallsprozess. Dieser wird durch die unterschiedlichen Werte im Parameter [`seed`] beeinflusst, aber es bleibt dennoch ein Zufallsprozess. Der Parameter [`seed`] ist somit kein Parameter, den man z.B. durch Vorwissen oder durch Simulationen auf den "besten"Wert festlegen kann. Aufgrund dieser Umstände durchläuft der Parameter [`seed`] bei jedem einzelnen Simulationsschritt eine gewisse Anzahl an Werten. Durch die dadurch gewonnene Simulationsstichprobengröße erhöht sich zusätzlich die Robustheit der aus den Simulationen gewonnenen Erkenntnisse. Um diese Robustheit entsprechend hoch und die Simulationsdauer entsprechend kurz zu halten, wurde die Anzahl der verschieden Werte für den Parameter [`seed`] auf 40 gesetzt.

- Ein wichtiger Punkt in der Simulationsstudie ist die benötigte Rechenzeit. Um diese möglichst gering zu halten, wurde der Parameter [`max.n.genes`] als erster Parameter ausgewählt, da hierbei aufgrund vorheriger Testversuche, die größte Zeitersparnis vermutet wurde. Die Default-Einstellung entspricht normalerweise der Anzahl der im Datensatz enthaltenen Gene. Da diese aber in diesem Fall die Anzahl der Beobachtungen übersteigen, wird sie auf die Anzahl der Beobachtungen zurückgestuft. Aufgrund der Festlegung des Datensatzes DKTK als Default-Einstellung, gilt folglich auch die Default-Einstellung [`max.n.genes = 85`]. Die restlichen Gene werden zuvor mit Hilfe univariater Modelle (siehe Abschnitt 3.2) aussortiert.

- Da ein Teil der Daten zur späteren Validierung mit dem c-Index benötigt wird, kann nicht der vollständige Datensatz für den Algorithmus verwendet werden (siehe hierzu Abschnitt 4.3.2). Wie bereits erwähnt, wird im Rahmen dieser Arbeit der Teil-Datensatz DKTK als Default-Einstellung verwendet und somit dem *rbsurv*-Algorithmus übergeben. Um diese Wahl zu überprüfen, ist der [`Datensatz`] der zweite Parameter, der in den Simulationen verändert wird. Der Datensatz für den Algorithmus sollte im Normalfall größer als 50% der Daten sein (siehe Kapitel 3.1). Für weitere Varianten des verwendeten Datensatzes wurden die Beobachtungen in verschiedene Verhältnisse aufgeteilt. Während die Aufteilung DKTK vs. KKG einem Verhältnis von ca. (52 : 48) Prozent entspricht, enthalten die anderen Varianten Verhältnisse von (50 : 50) bis zu (80 : 20) Prozent. Eine genaue Übersicht darüber befindet sich in Tabelle 5.

- Die verbleibenden Parameter wurden aufgrund möglicher Verkürzungen der Berechnungszeiten in folgender Reihenfolge verwendet: [`method`], [`n.iter`], [`n.fold`] und [`n.seq`].

Die letztlich verwendete Rangfolge ist in Tabelle 4 dargestellt. Zusätzlich sind auch die unterschiedlichen Ausprägungen vermerkt, die in den Simulationen benutzt wurden.

Rangfolge	Parameter	Ausprägungen
1	max.n.genes	10, 20, 30, 50, 85, 100
2	Datensatz	DKTK, T_60, T_70_1, T_70_2, T_80_1, T_80_2, 50, 60, 70, 80
3	method	efron, breslow, exact
4	n.iter	5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
5	n.fold	2, 3, 4, 5, 6, 7, 8, 9, 10
6	n.seq	1, 2, 3

Tabelle 4: Die Rangfolge und ihre Ausprägungen der Argumente im R-Paket *rbsurv* für die Simulationen

4.4.1 Hintergrund

Um bereits zu Beginn einen Eindruck von der Parameterwahl und ihren Auswirkungen zu bekommen, wurde beispielhaft der Parameter [`n.iter`] ausgewählt. Abbildung 6 zeigt den Unterschied für zwei Berechnungen mit unterschiedlichen Werten für den Parameter [`n.iter`] und die daraus resultierenden Variablen in den Modellen. Die Werte der restlichen Parameter entsprechen den Default-Einstellungen der *rbsurv*-Funktion. Hierbei lässt sich bereits gut erkennen, dass die Parameterwahl einen durchaus großen Einfluss auf die Variablen und damit auch auf die Güte eines Modelles hat. So findet sich in diesem Beispiel lediglich eine miRNA in beiden Modellen wieder. Im Anhang befindet sich die Tabelle 7, die den entsprechenden Namen der miRNA's zur ID-Nummer enthält. Im Folgenden werden die verschiedenen Iterationen durchgeführt, die auf Tabelle 4 basieren. Am Ende des Kapitels 4.4 befindet sich in Tabelle 6 zudem eine Übersicht über den Aufbau und die Ergebnisse der einzelnen Iterationen.

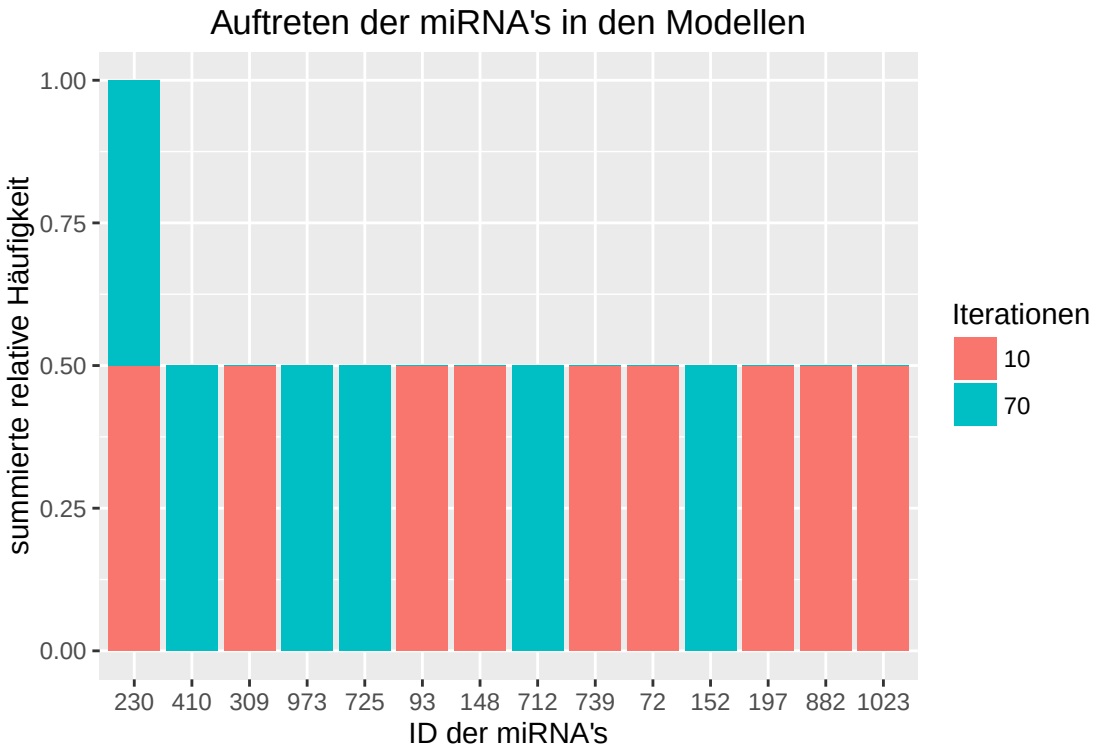

Abbildung 6: Die Auswirkungen der Anzahl der Iterationen auf die Variablenselektion. Die übrigen Parameter wurden auf die Default-Einstellungen gesetzt.

4.4.2 Iteration 1.1 bis 1.6

Wie bereits erwähnt, wird der Parameter [`seed`] in der Auswertung nicht auf einen festen Wert festgelegt. Stattdessen wird er immer wieder nach dem Zufallsprinzip neu bestimmt, um dadurch robustere Ergebnisse zu erhalten. Abbildung 7 zeigt allerdings, dass auch der Parameter [`seed`] einen relativ großen Einfluss auf die Güte eines Modelles haben kann. So schwankt allein in dieser kleinen Stichprobe von zehn verschiedenen seed's der c-Index um mehr als 0.12. Der c-Index wurde mit Hilfe des Validierungsdatensatzes (*hier:* KKG) berechnet. Allerdings ist der mittlere c-Index dieser zehn Berechnungen mit ca. 0.506 auch nur knapp über 0.5 (gestrichelte Linie) und deutet somit auf keine gute Modellanpassung hin.

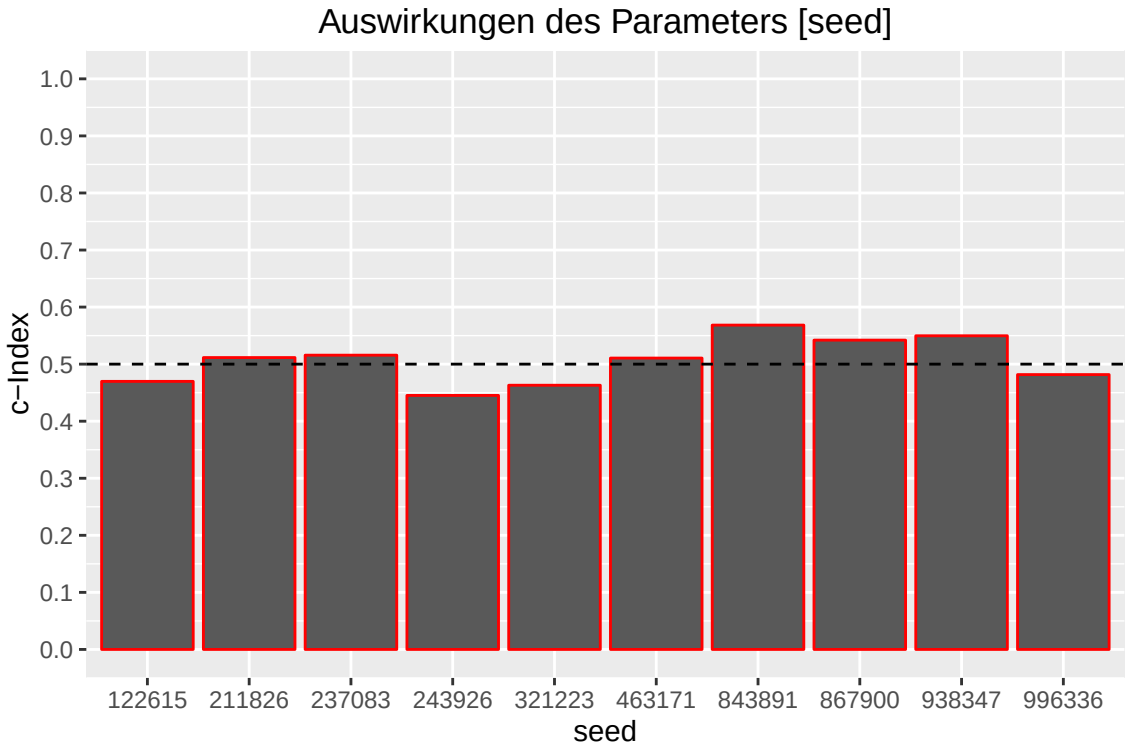

Abbildung 7: Die Auswirkungen von verschiedenen Werten des Parameters [`seed`] auf den c-Index. Die übrigen Parameter entsprechen den Default-Einstellungen.

- Iteration 1.1

  Im ersten Schritt zum optimalen Modell wird der Parameter [`max.n.genes`] festgelegt. Hierfür wurden die Default-Einstellungen der *rbsurv*-Funktion übernommen und kombiniert mit jeweils verschiedenen Werten für [`max.n.genes`]. Jede dieser Kombinationen wurde dann mit 40 verschiedenen Werten für den Parameter [`seed`] berechnet. Abbildung 8 zeigt den Vergleich der Berechnungen im Hinblick auf den c-Index. Hierbei lassen sich bereits deutliche Unterschiede erkennen. So schneidet die Default-Einstellung von 85 miRNA's mit einem durchschnittlichen c-Index von 0.516 relativ schlecht ab. Der durchschnittlich beste c-Index wurde dagegen mit der Einstellung [`max.n.genes = 20`] erreicht. Zusätzlich liegen hier auch die mittleren 50% der Daten am engsten beieinander und weisen somit die geringste Streuung in diesem Bereich auf.

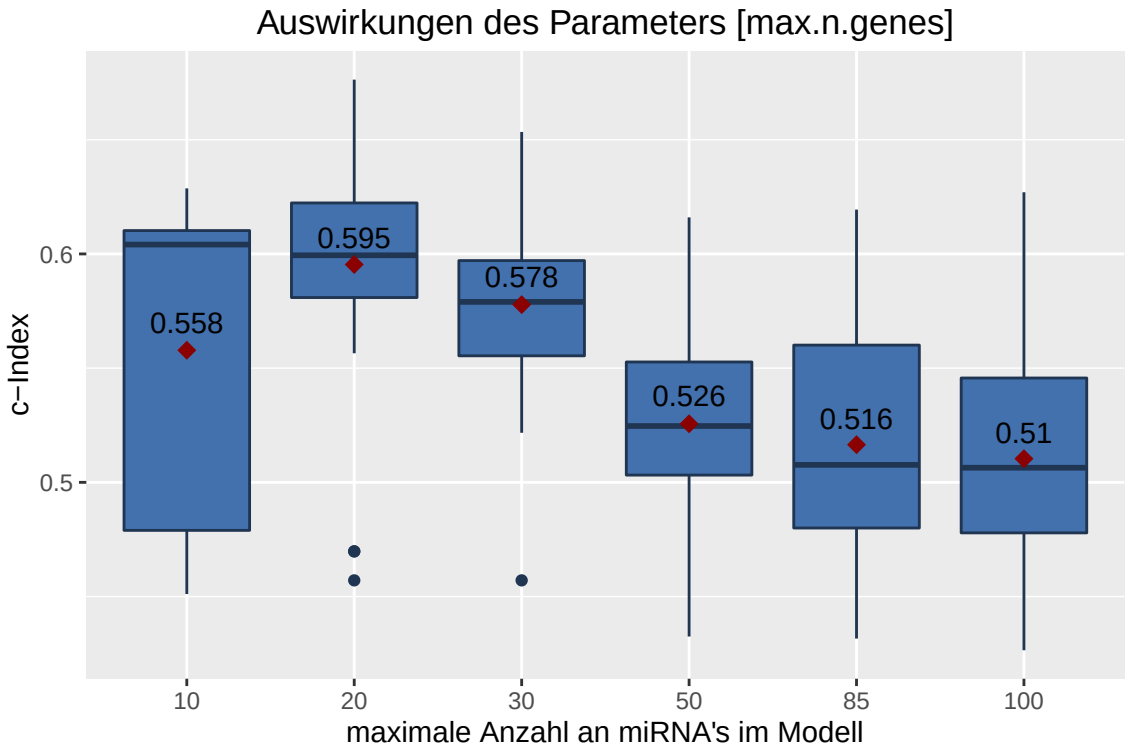

Abbildung 8: Die verschiedenen Werte des Parameters [`max.n.genes`] und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

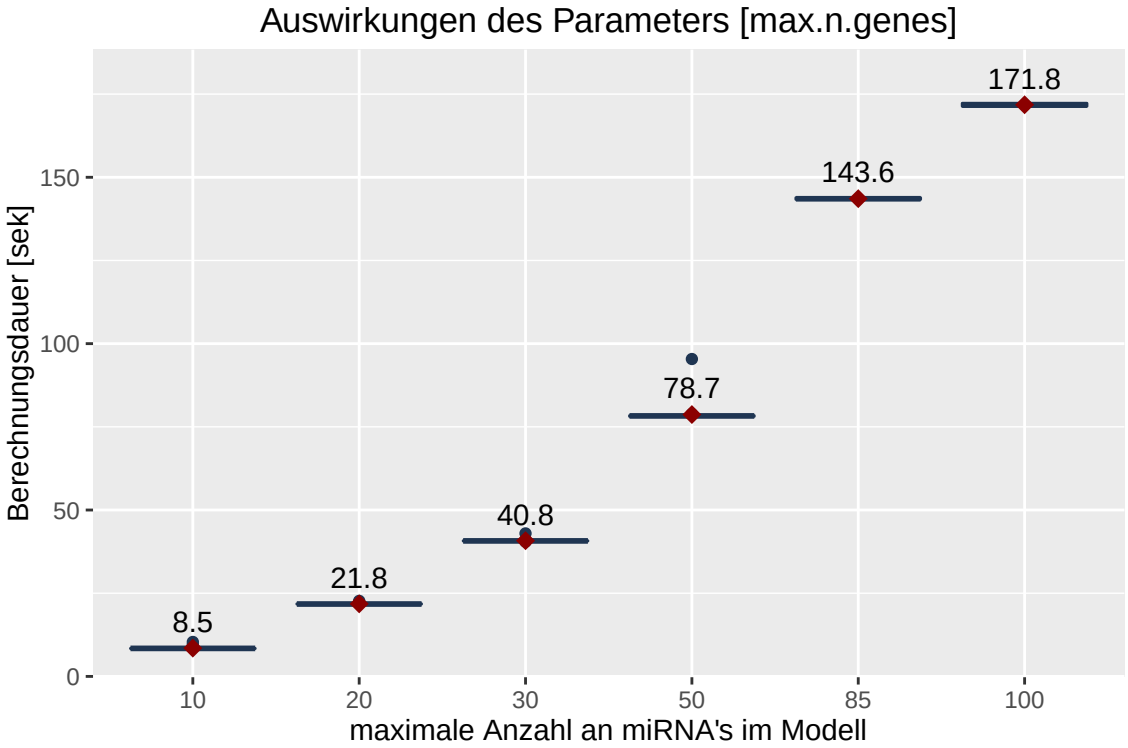

Abbildung 9: Die verschiedenen Werte des Parameters [`max.n.genes`] und ihre Auswirkungen auf die Berechnungsdauer. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Zudem lässt sich eine hohe positive Korrelation von ca. 0.999 (nach Pearson) zwischen der Berechnungsdauer und dem Parameter [`max.n.genes`] beobachten, wie auch in Abbildung 9 zu sehen ist. Aufgrund des deutlich besseren, da höheren c-Index und der relativ geringen Berechnungsdauer werden die weiteren Berechnungen mit der Parameterwahl [`max.n.genes = 20`] fortgeführt.

- Iteration 1.2

  Die zweite Iteration beinhaltet das Festlegen des Parameters [`Datensatz`]. Die bis hierhin verwendete Default-Einstellung war der Datensatz DKTK. Die ersten vier Teil-Datensätze wurden zufällig aus dem vollständigen Datensatz gebildet. Es wurde dabei also nicht auf die Gruppen- bzw. Institutszugehörigkeit geachtet. Bei den letzten sechs Teil-Datensätze wurde darauf geachtet, dass die einzelnen Institute mit ihren dazugehörigen Beobachtungen nicht getrennt wurden. Die weiteren getesteten Teil-Datensätze und ihre Zusammensetzung sind in Tabelle 5 zu finden.

Bezeichnung	Umfang (proz. Anteil)	enthaltene Datensätze
50	81 (50%)	zufällig gebildet aus vollständigem Datensatz
60	97 (60%)	zufällig gebildet aus vollständigem Datensatz
70	113 (70%)	zufällig gebildet aus vollständigem Datensatz
80	130 (80%)	zufällig gebildet aus vollständigem Datensatz
DKTK	85 (52%)	DD, FB, EU, FFM, TUE, TUM, BER, HD
T_60	97 (60%)	KKG, FB, TUM
T_70_1	113 (70%)	KKG, DD, FFM
T_70_2	113 (70%)	KKG, DD, TUE
T_80_1	130 (80%)	KKG, DD, FB, FFM
T_80_2	130 (80%)	KKG, DD, FB, TUE

Tabelle 5: Die verwendeten Trainings-Datensätze und ihre Zusammenstellung

Betrachtet man die Auswirkung der verschiedenen Teil-Datensätze zur Modellwahl in Abbildung 10, so sind durchaus Unterschiede erkennbar. Der c-Index der jeweiligen Teil-Datensätze unterscheidet sich dabei sowohl im Mittelwert als auch in der Streuung deutlich. Dabei ist aber keine Tendenz auszumachen, dass ein größerer Teil-Datensatz zu einem besseren Ergebnis führen würde. Da sich die jeweiligen Berechnungszeiten der Teil-Datensätze nicht beträchtlich voneinander unterscheiden (siehe Anhang: Abbildung 34), wird der Parameter [`Datensatz`] weiterhin auf *DKTK* gesetzt. Zwar weist der Teildatensatz *60* einen minimal höheren mittleren c-Index auf, allerdings ist die Bildung dieses Teildatensatzes sehr vom Zufall geprägt und dementsprechend weniger aussagekräftig.

Im Vergleich des c-Index der beiden Teildatensätze *60* und *DKTK* ist also kein großer Unterschied zu bemerken. Schauen wir uns allerdings das relative Vorkommen der miRNA's in den Modellen an, so basieren die jeweiligen Modelle fast ausschließlich auf unterschiedlichen Variablen. In Abbildung 11 sind die 15 am häufigsten vorkommenden miRNA's zu sehen. Lediglich die Nummer 973 ist in beiden Modellgruppen sehr häufig zu finden. Ansonsten gibt es fast keine Überschneidungen. Aufgrund der Übersichtlichkeit befinden sich auch in den weiteren Abbildungen des selben Musters lediglich die 15 am häufigsten vorkommenden miRNA's. Die vollständigen Abbildungen befinden sich im digitalen Anhang.

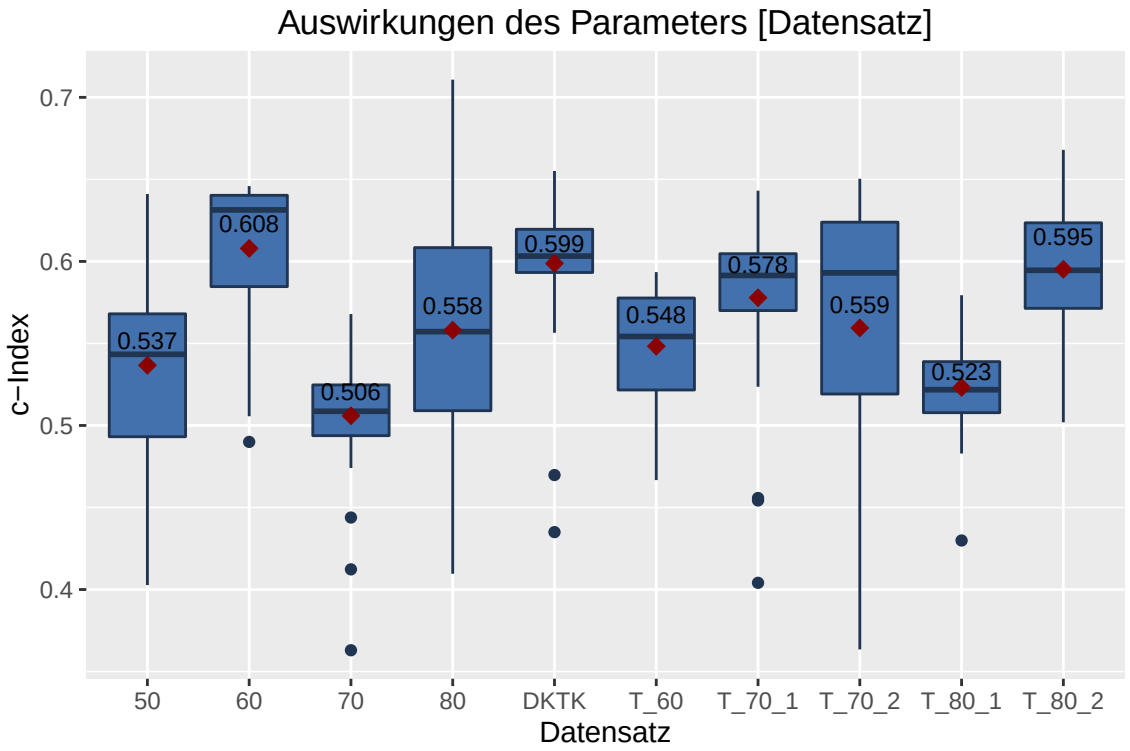

Abbildung 10: Die verschiedenen Trainings-Datensätze und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

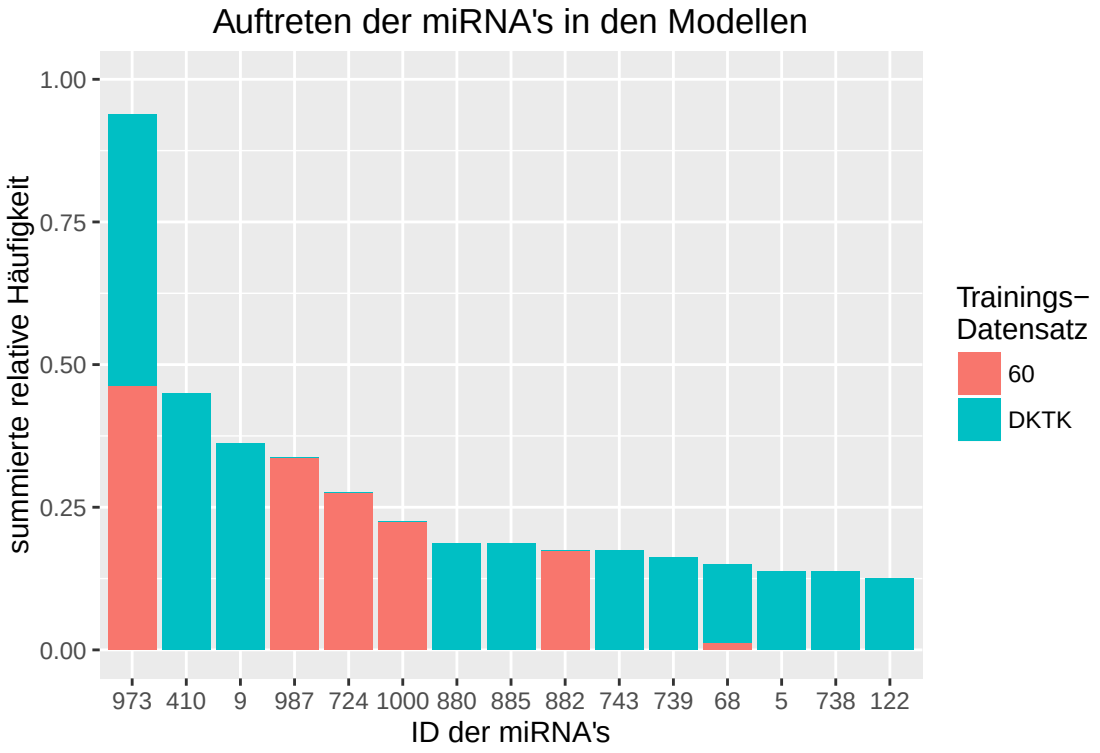

Abbildung 11: Vergleich zweier Teil-Datensätze und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's in den zugehörigen Modellen. Es wurden pro Teil-Datensatz 40 Modelle berechnet.

- Iteration 1.3

  In Iteration 1.3 geht es um die verwendete Berechnungsmethode bei Bindungen. Hierfür stellt das Paket drei verschiedene Optionen bereit. Zusätzlich zur Default-Einstellung [`method = efron`] gibt es noch die exakte Berechnung nach Kalbfleisch und Prentice (2002) und die Methode nach Breslow (1974). Vergleichen wir die drei Methoden im Hinblick auf den c-Index (siehe Abbildung 12), so sind keine großen Unterschiede zu bemerken. Dies lässt sich auch damit begründen, dass die Daten insgesamt relativ wenig Bindungen enthalten. Wären überhaupt keine Bindungen vorhanden, so würden sich für alle drei Methoden die gleichen Ergebnisse ergeben.

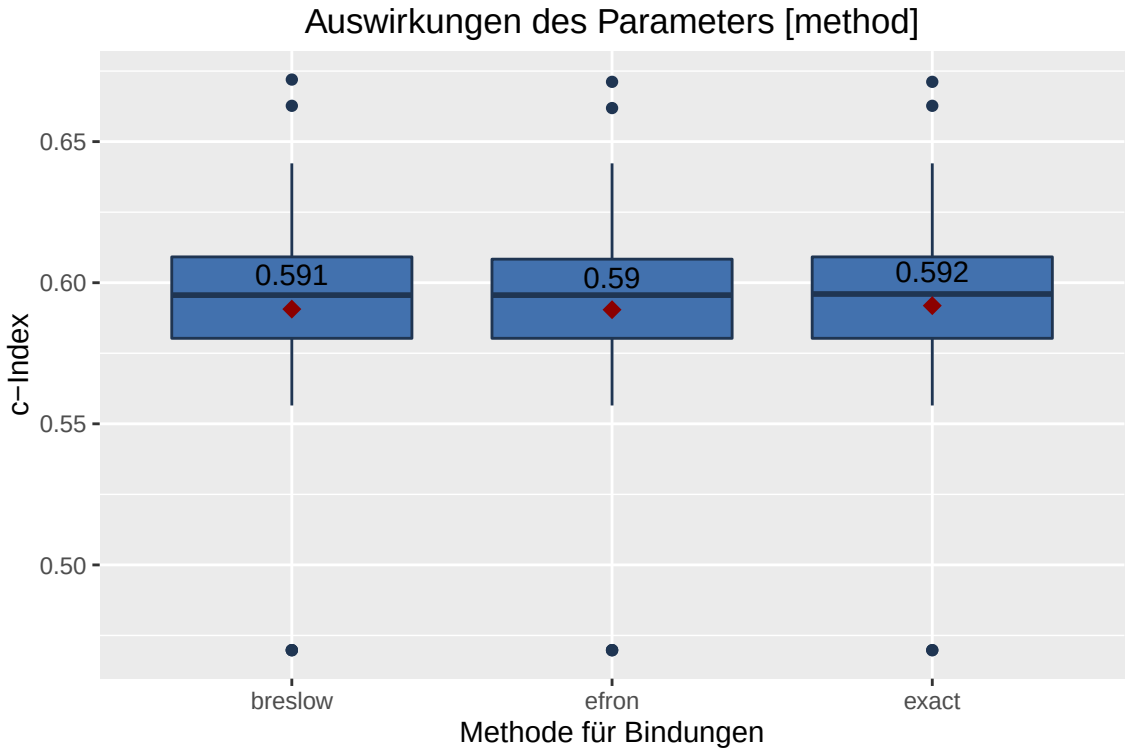

Abbildung 12: Die verschiedenen Methoden für die Berechnung der Likelihood und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Die Ursache für die geringen Unterschiede im c-Index rühren daher, dass sich die Modelle kaum unterscheiden. Bei der Betrachtung der enthaltenen miRNA's in Abbildung 13 fällt auf, dass die drei Modellgruppen fast immer zu gleichen Anteilen die miRNA's enthalten. Da sich die verschiedenen Methoden im vorhandenen Datensatz kaum unterschiedlich bemerkbar machen, wird vorerst die Default-Einstellung [`method = efron`] beibehalten.

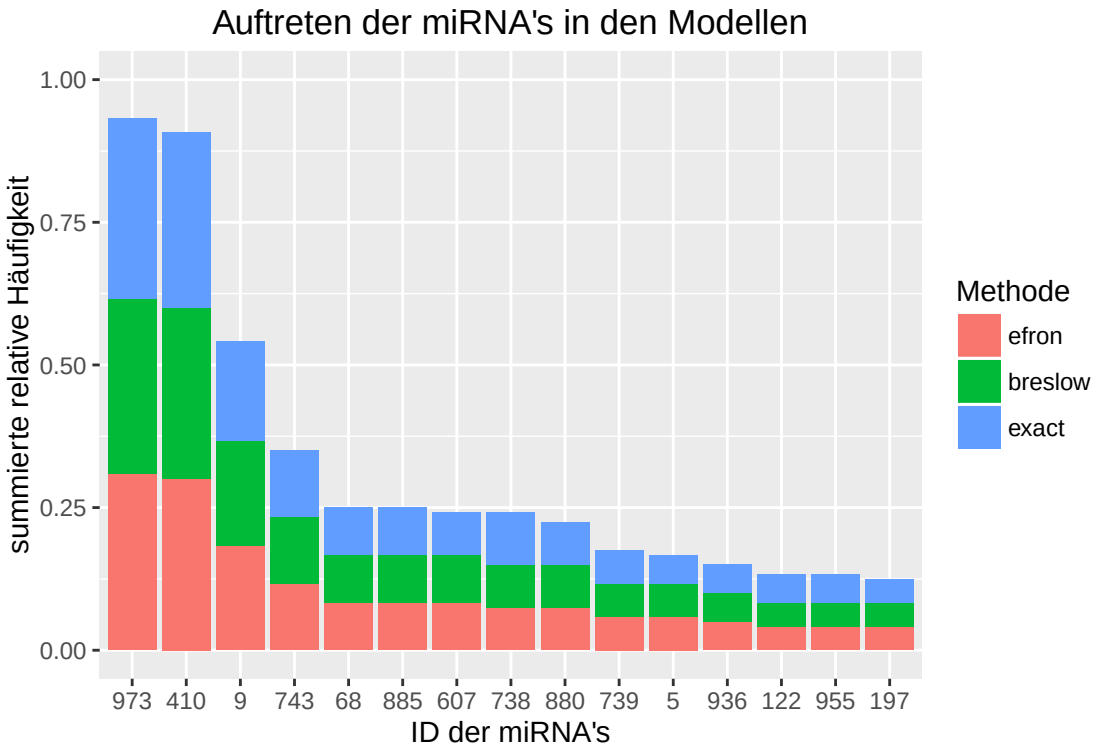

Abbildung 13: Vergleich dreier Methoden für die Berechnung der Likelihood bei vorhandenen Bindungen und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's in den zugehörigen Modellen. Es wurden pro Methode 40 Modelle berechnet.

- Iteration 1.4

  Der Parameter [`n.iter`] dient in der *rbsurv*-Funktion dazu robustere Schätzungen für die berechneten Parameter zu erlangen. Bei Betrachtung der Abbildung 14 mit verschiedenen Werten für diesen Parameter, kann keine positive Korrelation erkannt werden. Tatsächlich ist die Korrelation des Parameter [`n.iter`] und des c-Index mit -0.19 (Pearson) leicht negativ, was somit für eine geringe Anzahl an Iterationen spricht. Auffällig ist zudem, dass die Größe der Boxen im Boxplot mit zunehmender Anzahl an Iterationen ebenfalls ansteigt. Dies spricht dafür, dass Modelle, welche mit einer hohen Anzahl an Iterationen berechnet wurden, eine höhere Schwankung der Modellgüte vorweisen.

  Um festzustellen, welche Auswirkungen der Parameter auf die Variablenselektion hat, schauen wir uns zusätzlich Abbildung 15 an. Die drei repräsentativen Werte für den Parameter zeigen, dass relativ viele miRNA's in allen Modellgruppen zumindest zum Teil vorkommen. Die Modelle setzen sich also vermutlich zum Großteil aus demselben Pool an miRNA's zusammen, allerdings ist die Zusammenstellung der einzelnen miRNA's bei Modellen mit hoher Anzahl an Iterationen im Durchschnitt schlechter.

  Der hohe Anstieg der Berechnungsdauer (siehe Anhang: Abbildung 35) und der zusätzlich beste mittlere c-Index sprechen dafür, den Parameter [`n.iter`] auf fünf Iterationen festzulegen.

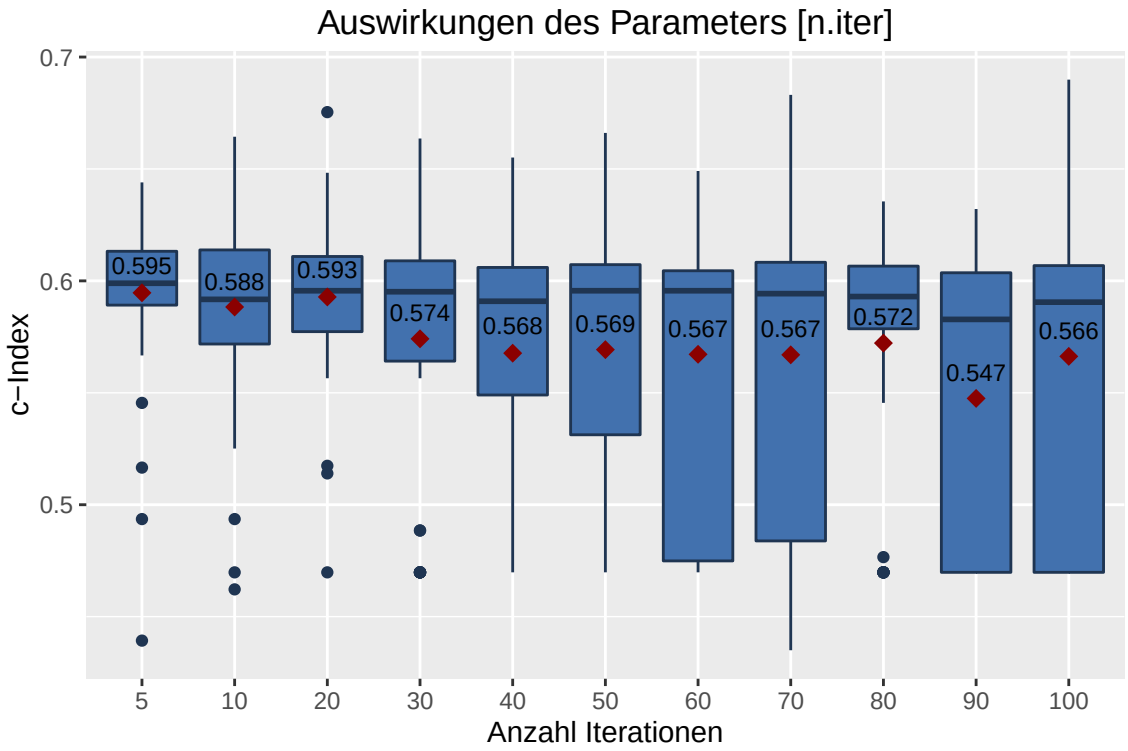

Abbildung 14: Die verschiedene Anzahl an Iterationen in der *rbsurv*-Funktion und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

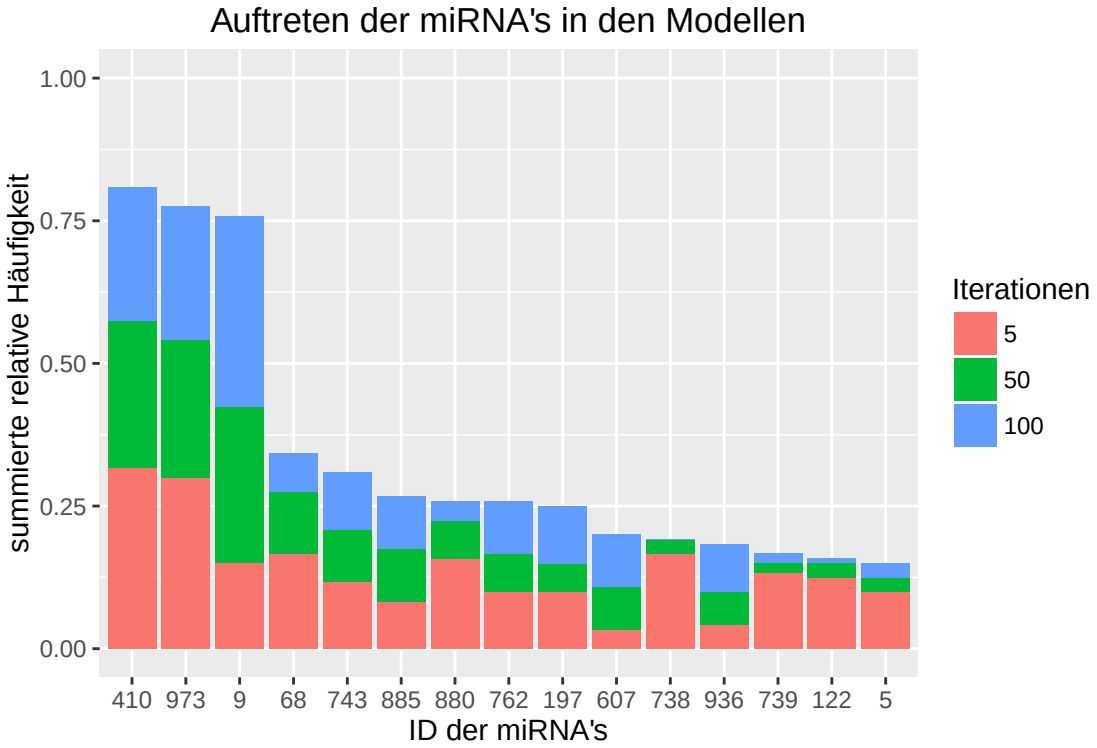

Abbildung 15: Vergleich dreier verschiedener Anzahlen an Iterationen in der *rbsurv*-Funktion und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's in den zugehörigen Modellen. Es wurden pro Wert 40 Modelle berechnet.

- Iteration 1.5

  Die fünfte Iteration behandelt den Parameter [`n.fold`]. Dessen Default-Einstellung liegt hierbei bei [`n.fold = 3`]. Das bedeutet, dass innerhalb der *rbsurv*-Funktion eine Einteilung der Daten zu zwei Drittel zum Trainingsdatensatz und zu einem Drittel zum Validierungsdatensatz erfolgt. Diese Einteilung erfolgt ausschließlich im Algorithmus der Funktion und hat nichts mit der anschließenden Validierung zur Gewinnung des c-Index zu tun. Diese basiert auf den Daten, die nicht im Algorithmus verwendet wurden.
  Setzt man den Parameter beispielsweise auf [`n.fold = 5`], so teilt der Algorithmus die Daten so ein, dass der Trainingsdatensatz vier Fünftel und der Validierungsdatensatz ein Fünftel der Daten enthält. Diese Einteilung wird mehrmals gemäß der Einstellung des Parameters [`n.iter`] durchgeführt (siehe Abschnitt 3.2).

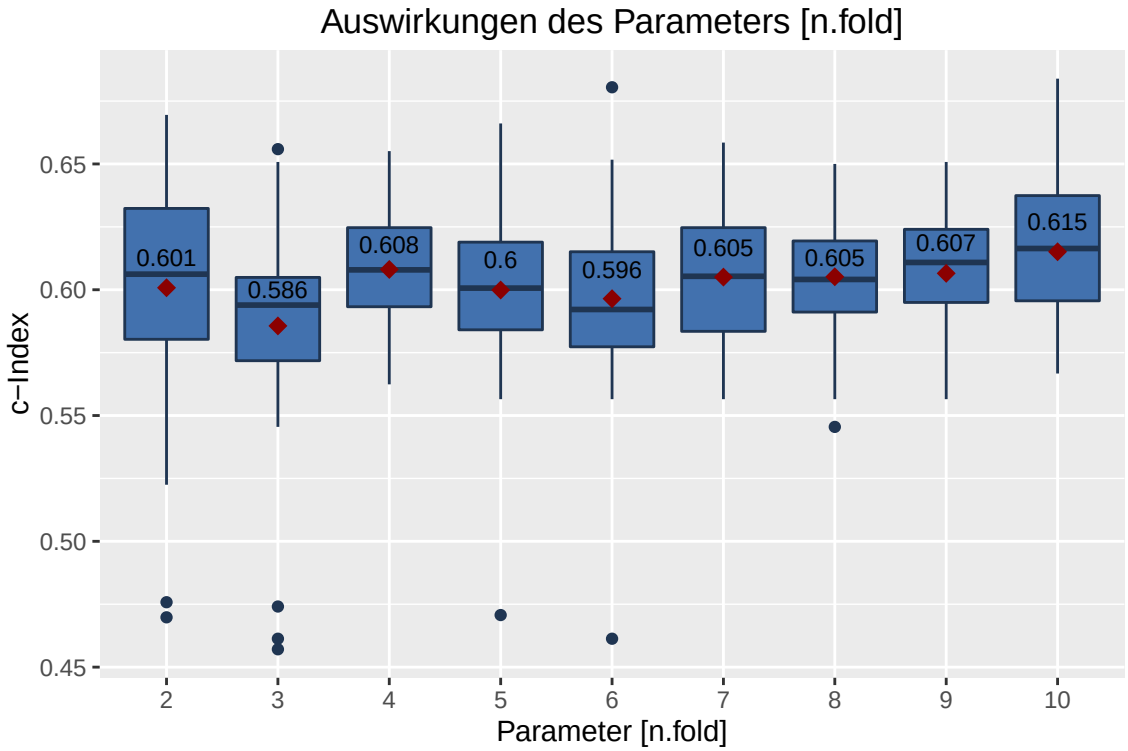

Abbildung 16: Die unterschiedlichen Werte des Parameters [`n.fold`] in der *rbsurv*-Funktion und der dazugehörige c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

In Abbildung 16 ist der c-Index für neun verschiedene Werte des Parameters [`n.fold`] abgebildet. Dabei entsprechen die Einstellungen [`n.fold = 2`] und [`n.fold = 10`] dem Minimum bzw. dem Maximum der möglichen Werte. Damit auch der Validierungsdatensatz ausreichend Informationsgehalt besitzt, achtet der Algorithmus darauf, dass die folgende Formel eingehalten wird:

$$n_1 > 3 * [nfold], \tag{11}$$

mit
n_1 = Summe der Beobachtungen im Datensatz mit Status = 1 (Cho et al., 2009).

Da für den Teil-Datensatz DKTK $n_1 = 32$ gilt, ist die größte ganze Zahl, welche die Gleichung erfüllt, die 10. Für das Minimum des Parameters ist die 2 zudem eine sinnvolle Wahl, da somit gewährleistet ist, dass mindestens die Hälfte der Daten als Trainingsdatensatz genutzt werden können. Werden die möglichen Einstellungen für den Parameter [`n.fold`] im Blick auf den c-Index in Abbildung 16 betrachtet, so fällt auf, dass die Default-Einstellung von [`n.fold = 3`] im Mittel am schlechtesten abschneidet. Der beste mittlere c-Index berechnet sich auf der Grundlage [`n.fold = 10`], was bedeutet, dass der Trainingsdatensatz 90% der verwendeten Daten enthält. Allerdings liegen die Mittelwerte alle relativ eng beieinander und es ist keine klare Tendenz zu erkennen. Zudem besitzt der Parameter [`n.fold`] keinen großen Einfluss auf die Berechnungszeit (siehe Anhang: Abbildung 36) und wird deshalb auf [`n.fold = 10`] gesetzt.

- Iteration 1.6

  Der letzte Parameter im ersten Durchlauf ist der Parameter [`n.seq`], der dazu dient multiple Modelle zu bilden. Setzt man den Parameter beispielsweise auf [`n.seq = 3`], so bildet der Algorithmus drei eigenständige Modelle, die insofern voneinander abhängig sind, dass sie keine Überschneidungen in den enthaltenen miRNA's besitzen. Zur besseren Vergleichbarkeit wurden im Zuge dieser Auswertung bei der Bildung multipler Modelle anschließend alle ausgewählten Variablen in ein Modell übergeben. Dies kann dazu führen, dass die Modelle statistisch gesehen nicht optimal sind, da es zwischen den Einflussvariablen durchaus hohe Korrelationen geben kann. Allerdings können sie dadurch, aus biologischer Sicht, mehr wichtige Einflussvariablen enthalten, die sonst eventuell nicht beachtet worden wären.

  Aus Gründen der Berechnungszeit sind in dieser Auswertung lediglich maximal drei verschiedene Modelle berechnet worden. In Abbildung 17 sind diese Modellgruppen zu sehen. Hierbei ist die Einstellung [`n.seq = 3`] die mit dem besten mittleren c-Index. Betrachten wir dazu die verwendete Berechnungszeit in Abbildung 18, so lässt sich auch hier ein großer Zusammenhang erkennen (Korrelationskoeffizient nach Pearson: 0.874).

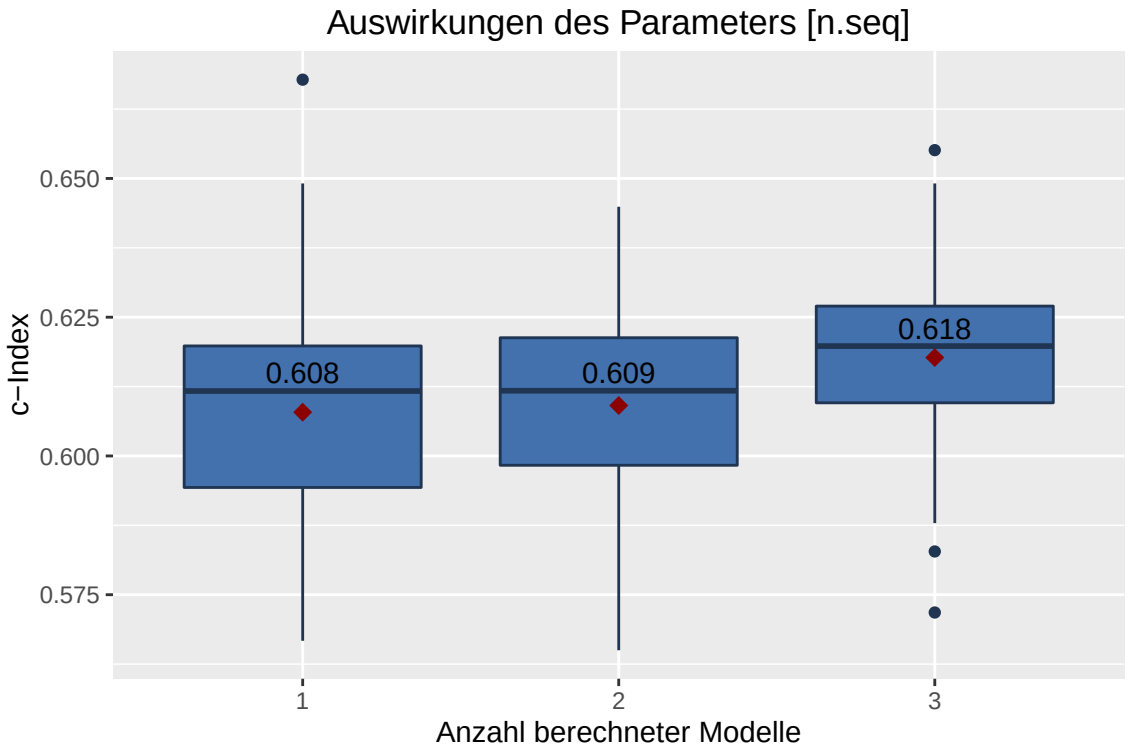

Abbildung 17: Unterschiedliche Anzahlen an multiplen Modellen und der dazugehörige c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

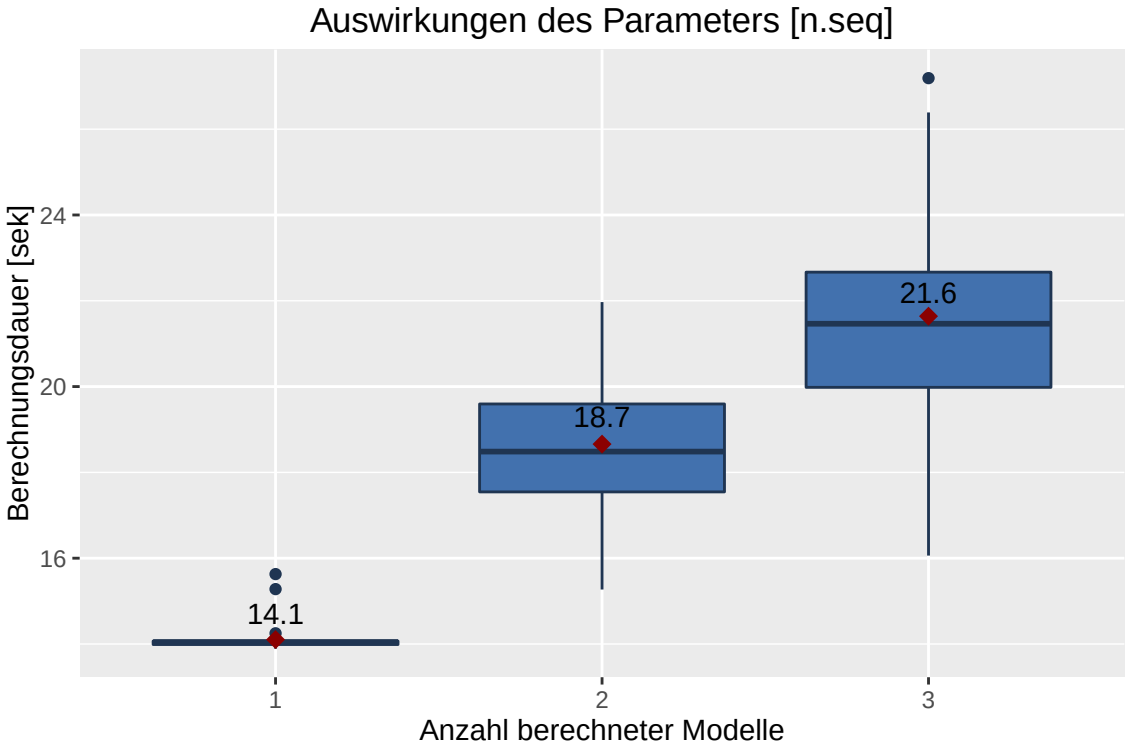

Abbildung 18: Unterschiedliche Anzahlen an multiplen Modellen und ihre Auswirkungen auf die Berechnungsdauer. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Abbildung 19 zeigt die verwendeten miRNA's in den Modellen. Hier lässt sich kein großer Unterschied erkennen, was auch darin begründet ist, dass bei multiplen Modellen lediglich neue miRNA's zum bereits vorhandenen Modell hinzugefügt werden. Dadurch kann der Anteil an auftretenden miRNA's in den Modellen mit zunehmender Anzahl an multiplen Modellen ebenfalls nur zunehmen oder gleich bleiben.

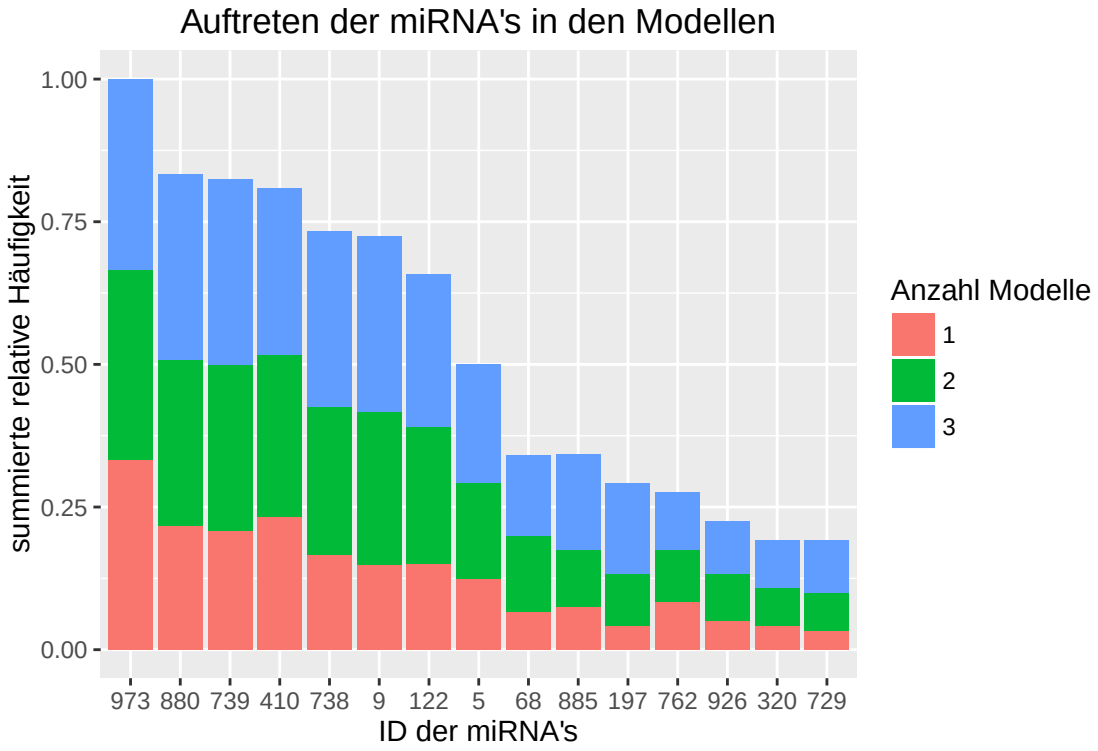

Abbildung 19: Vergleich dreier verschiedener Anzahlen an multiplen Modellen in der *rbsurv*-Funktion und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's. Es wurden pro Gruppe 40 Modelle berechnet.

Obwohl der mittlere c-Index bei drei Modellberechnungen am höchsten ist, wird der Parameter aufgrund der Berechnungszeit auf [`n.seq = 1`] belassen. Um allerdings ein optimales Modell mit hoher Prognosegüte zu erhalten, wäre es durchaus sinnvoll auch multiple Modelle zu verwenden.

4.4.3 Iteration 2.1 bis 2.6

Der zweite Durchgang optimiert die einzelnen Parameter in der gleichen Reihenfolge wie im ersten Durchgang. Die im ersten Durchgang bereits optimierten Parameter werden weiterhin als neue Default-Einstellungen verwendet. Um eine bessere Vergleichbarkeit zwischen den zwei Durchgängen zu ermöglichen, wurden innerhalb der Parameter immer die selben `seed's` verwendet. Das bedeutet, dass sich sowohl in Iteration 1.1 als auch in Iteration 2.1 dieselben `seed's` finden. Das gleiche gilt für Iteration 1.2 und Iteration 2.2, usw. .

- Iteration 2.1

  In der ersten Iteration des zweiten Durchgangs wird der Parameter [`max.n.genes`] zum zweiten Mal optimiert. Bei der Betrachtung des c-Index für verschiedene maximale Anzahlen an miRNA's in den Modellen in Abbildung 20 lässt sich ein ähnlicher Verlauf wie im ersten Durchgang (siehe Abbildung 8) erkennen. Es fällt zudem auf, dass der mittlere c-Index für jeden Wert des Parameters [`max.n.genes`] gestiegen ist. Dies ist auf die im ersten Durchgang optimierten Parameter zurückzuführen.

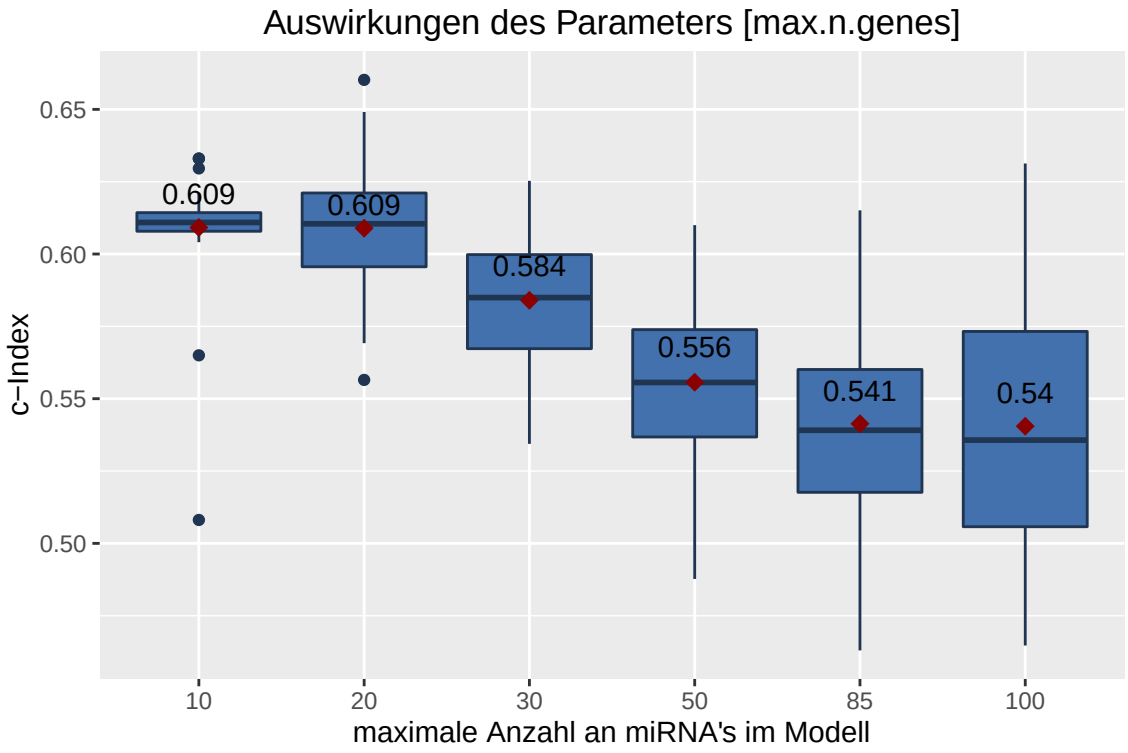

Abbildung 20: Die verschiedenen Werte des Parameters [`max.n.genes`] und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Der höchste mittlere c-Index wird hier durch Werte von zehn bzw. 20 erreicht. Aufgrund dessen, dass eine Beschränkung von über 1000 miRNA's auf nur maximal zehn miRNA's, die in das Modell mitaufgenommen werden können, relativ stark eingrenzt, wird der Parameter weiterhin auf [`max.n.genes = 20`] festgelegt.

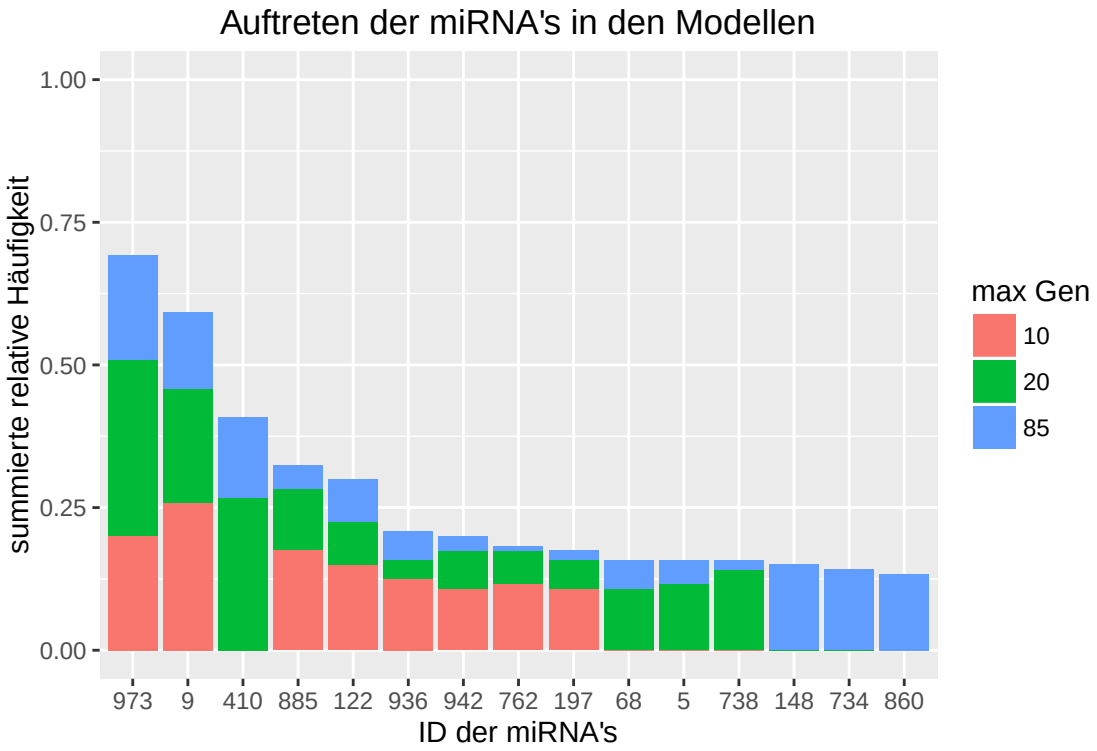

Abbildung 21: Vergleich dreier Anzahlen an maximalen miRNA's in der *rbsurv*-Funktion und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's. Es wurden pro Gruppe 40 Modelle berechnet.

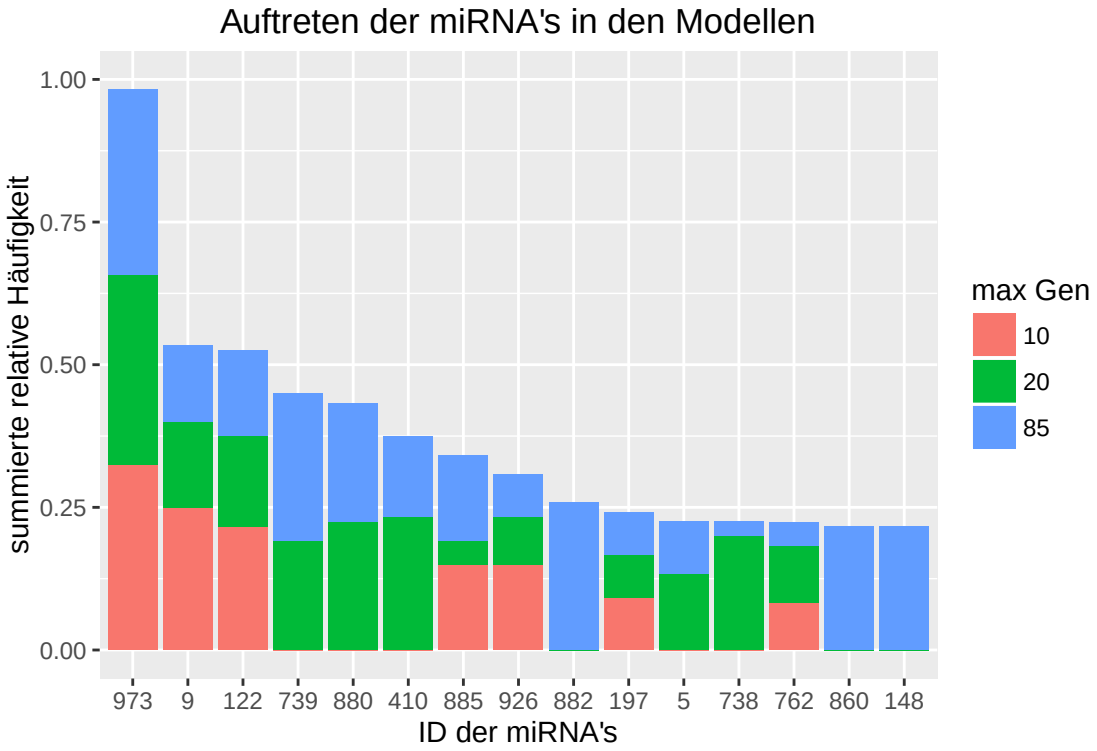

Abbildung 22: Vergleich dreier Anzahlen an maximalen miRNA's in der *rbsurv*-Funktion und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's. Es wurden pro Gruppe 40 Modelle berechnet.

In Abbildung 21 und 22 ist ein Vergleich der Variablenselektionen zwischen dem ersten und zweiten Durchgang zu sehen. Hierbei wurden jeweils verschiedene Werte für den Parameter [`max.n.genes`] betrachtet und die dazugehörigen Variablen, die dadurch in die Modelle mit aufgenommen wurden. Elf von den 15 am häufigsten vorgekommenen miRNA's im ersten Durchgang finden sich auch im zweiten Durchgang unter den Top 15 der miRNA's wieder. Ebenfalls auffällig ist das Ansteigen der summierten relativen Häufigkeit. Zwar steigt die summierte relative Häufigkeit nicht für jede miRNA an, allerdings ist das Niveau insgesamt höher. Die vorkommenden miRNA's im zweiten Durchgang treten dementsprechend öfters vermehrt in den Modellen auf als noch im ersten Durchgang. So kommt beispielsweise im ersten Durchgang die am 15. häufigste auftretende miRNA mit der ID 860 auf 16 Modelle, in welchen sie auftritt. Im zweiten Durchgang ist die am 15. häufigste auftretende miRNA (ID: 148) in 26 Modellen vertreten. Dieser Vergleich zeigt auf, dass durch die optimierten Parameter häufiger dieselben und vermutlich auch wichtigeren miRNA's für die Modelle ausgewählt werden und die Variabilität der Variablenselektion dadurch geringer wird.

- Iteration 2.2

  Der Parameter [`Datensatz`] wurde im ersten Durchgang nicht verändert und ist unverändert auf die Default-Einstellung *DKTK* eingestellt. Die Simulationen im zweiten Durchgang zeigen jedoch deutliche Veränderungen gegenüber dem ersten Durchgang. Bei der Betrachtung des mittleren c-Index in Abbildung 23 lässt sich gut erkennen, dass der Datensatz *T_80_2* den höchsten c-Index besitzt. Besonders der Unterschied zwischen dem Datensatz *T_80_2* und dem zufällig gebildeten Datensatz *80* ist auffällig. Obwohl beide Datensätze jeweils 80% der gesamten Daten enthalten, weisen sie doch deutliche Unterschiede im mittleren c-Index auf. Allerdings kann man daraus keine allgemeine Aussage ableiten, da dies lediglich auf diese zwei Datensätze zutrifft. Der Datensatz *60*, der noch im ersten Durchgang den höchsten c-Index aufweisen konnte (siehe Abbildung 10), ist dagegen lediglich im Mittelfeld der hier aufgeführten Datensätze zu finden. Die Berechnungsdauer hat keine entscheidende Bedeutung (siehe Anhang: Abbildung 37). Für die weiteren Simulationen wurde die Einstellung [`Datensatz = T_80_2`] übernommen.

- Iteration 2.3

  In Iteration 2.3 wird der Parameter [`method`] nochmals genauer betrachtet. Abbildung 24 zeigt hierbei ein ähnliches Bild wie bereits im ersten Durchgang (siehe Abbildung 12). So haben die unterschiedlichen Methoden für die Berechnung der partiellen Likelihood hier keinen großen Einfluss auf den c-Index. Die Parametereinstellung [`method = exact`] weist zwar einen etwas höheren Median für den c-Index auf, aber die Mittelwerte sind jeweils fast identisch. Allerdings ist zu beobachten, dass auch hier der mittlere c-Index durch die optimierten Parameter im Vergleich zum ersten Durchgang angestiegen ist.

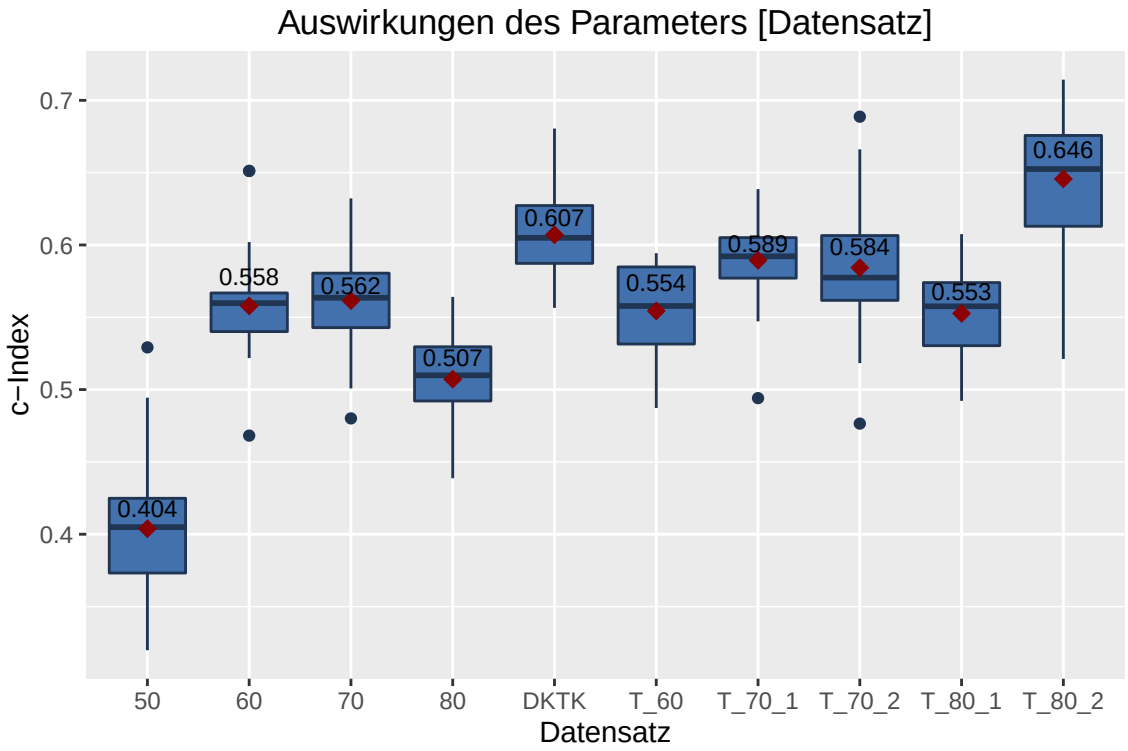

Abbildung 23: Die verschiedenen Trainings-Datensätze und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Vergleicht man allerdings die enthaltenen miRNA's in den Modellen, so ergeben sich große Unterschiede. In Abbildung 25 sind die 15 am häufigsten vorkommenden miRNA's in den Modellen zum Parameter [`method`] zu sehen. Vergleicht man diese mit Abbildung 13, so sind bis auf die miRNA mit der ID 973 keine Gemeinsamkeiten zu erkennen. Auch die vollständige Auflistung aller in den Modellen vorkommenden miRNA's liefert dasselbe Ergebnis (siehe digitaler Anhang). Das bedeutet, dass die gebildeten Modelle im ersten und im zweiten Durchgang bis auf die miRNA mit der ID 973 völlig unterschiedliche Einflussvariablen besitzen. Dieser große Unterschied ist vermutlich damit zu begründen, dass in der vorherigen Iteration 2.2 der Trainings-Datensatz geändert wurde. Die Verhältnisse zwischen den unterschiedlichen Methoden und dem Vorkommen der einzelnen miRNA's scheinen relativ konstant proportional zueinander zu sein und bestätigen somit den Eindruck aus Abbildung 25, dass die Methode für Bindungen hier keinen großen Einfluss auf die Modelle hat. Dadurch wird auch im weiteren Verlauf der Parameter auf der Default-Einstellung [`method = efron`] belassen.

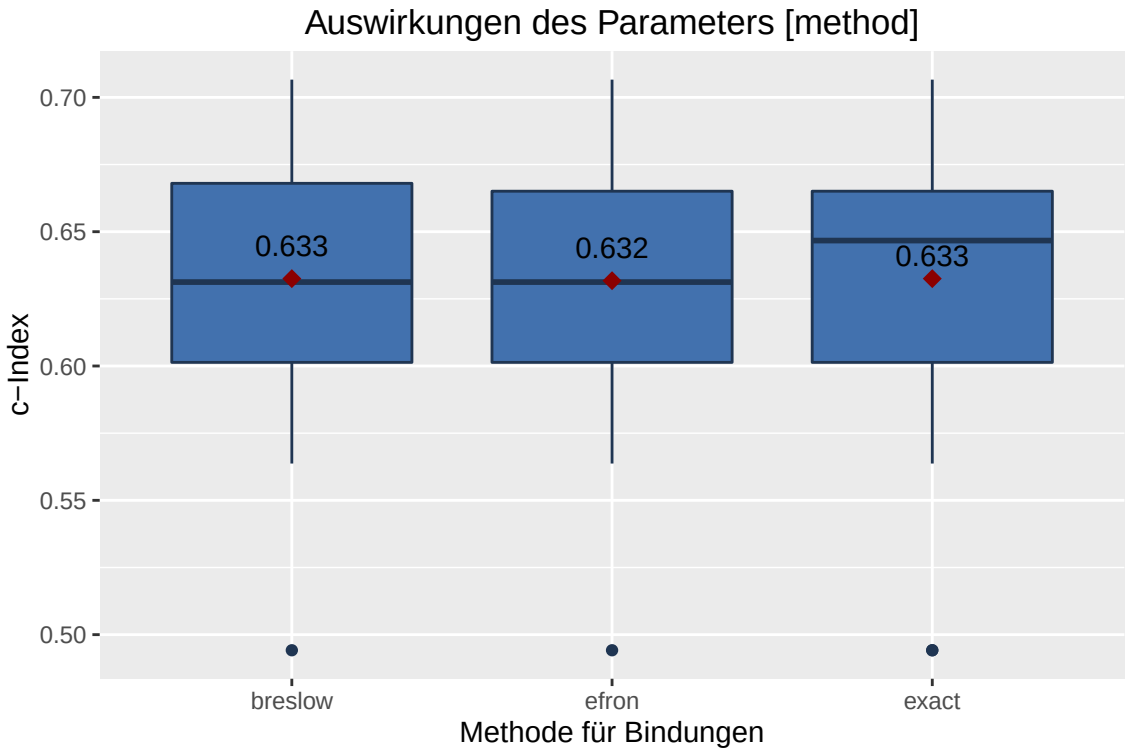

Abbildung 24: Die verschiedenen Methoden für die Berechnung der Likelihood und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

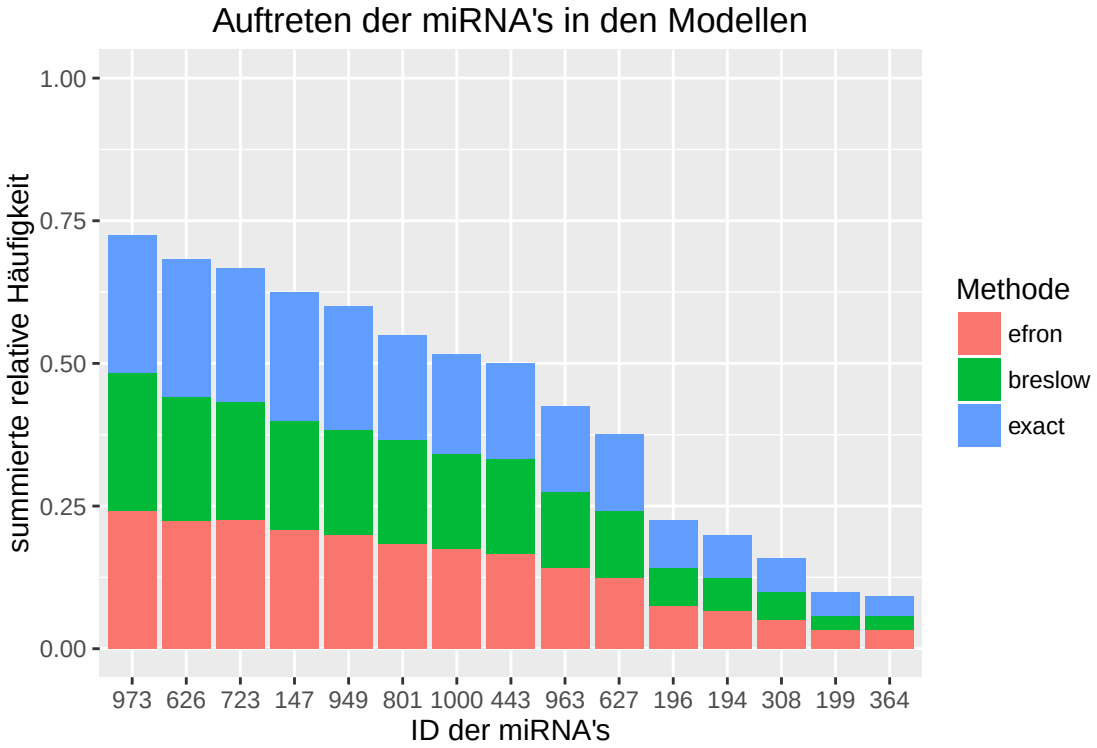

Abbildung 25: Vergleich dreier Methoden für die Berechnung der Likelihood bei vorhandenen Bindungen und das relative Vorkommen der 15 am häufigsten auftretenden miRNA's in den zugehörigen Modellen. Es wurden pro Methode 40 Modelle berechnet.

- Iteration 2.4

  Der Parameter [`n.iter`] wurde im ersten Durchgang von der Default-Einstellung [`n.iter = 10`] auf [`n.iter = 5`] optimiert. Durch Abbildung 26 bestätigt sich diese Optimierung auch im zweiten Durchgang, da auch hier die Einstellung [`n.iter = 5`] den höchsten mittleren c-Index aufweist. Der Korrelationskoeffizient des Parameter [`n.iter`] und des c-Index ist mit -0.19 (Pearson) leicht negativ und entspricht ziemlich exakt dem gleichen Korrelationskoeffizienten wie noch im ersten Durchgang. Die Berechnungsdauern, die in Abbildung 27 zu finden sind, steigen proportional zur Anzahl der Iterationen deutlich an. So ist auch hier eine geringe Anzahl an Iterationen im Algorithmus zu befürworten und dementsprechend wird auch im zweiten Durchgang der Parameter auf [`n.iter = 5`] festgelegt.

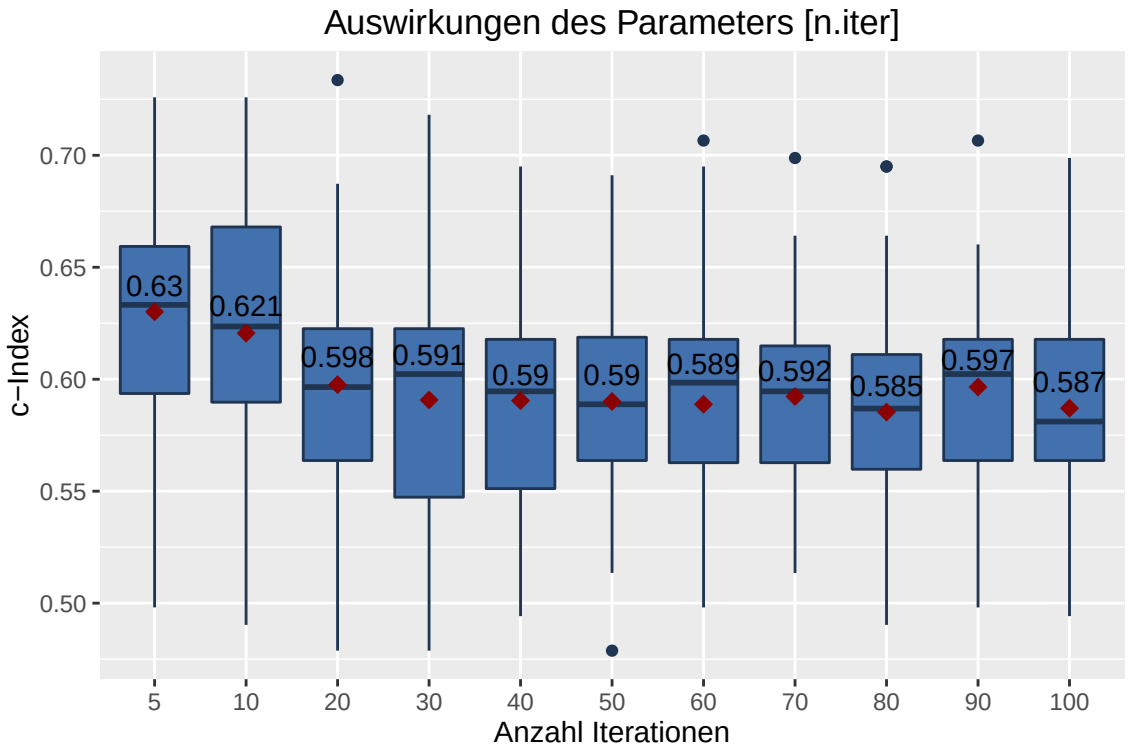

Abbildung 26: Die verschiedene Anzahl an Iterationen in der *rbsurv*-Funktion und ihre Auswirkungen auf den c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

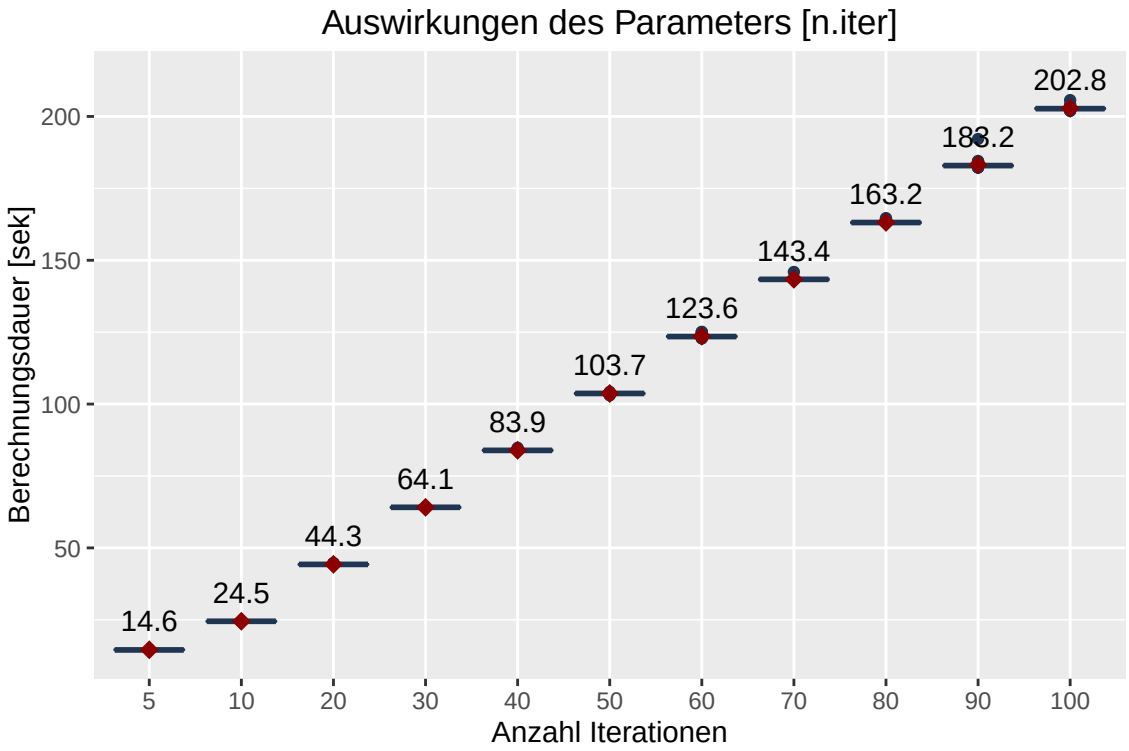

Abbildung 27: Die verschiedene Anzahl an Iterationen in der *rbsurv*-Funktion und ihre Auswirkungen auf die Berechnungsdauer. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

- Iteration 2.5

  Um den Parameter [`n.fold`] ein weiteres Mal zu optimieren, wurden dieselben Werte für den Parameter simuliert wie bereits in Iteration 1.5. Der Überblick über den jeweils dazugehörigen mittleren c-Index befindet sich in Abbildung 28. Auch hier bestätigt sich der Eindruck aus dem ersten Durchgang, dass sich ein hoher Wert für den Parameter [`n.fold`] positiv auf den c-Index auswirkt.

  Die Berechnungsdauer ist für jeden der einzelnen Werte so gut wie identisch (siehe Anhang: Abbildung 38) und spielt somit keine Rolle für die Optimierung des Parameters. Aufgrund des höchsten c-Index wird dieser weiterhin auf [`n.fold = 10`] gesetzt.

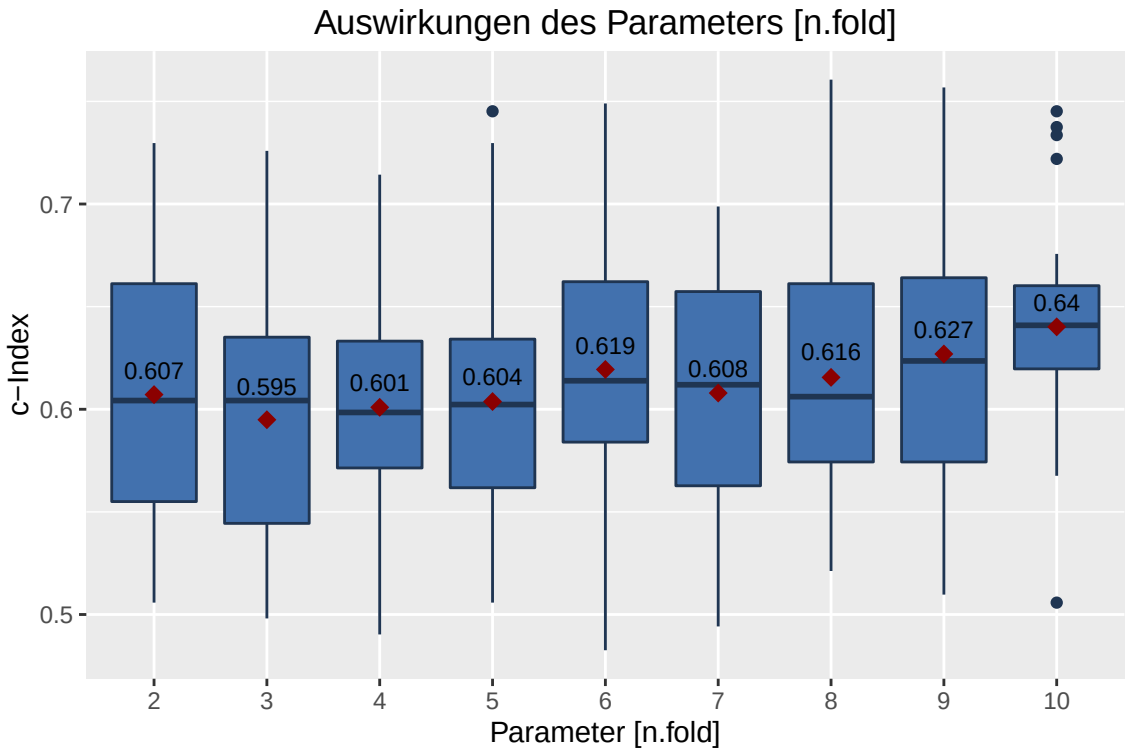

Abbildung 28: Die unterschiedlichen Werte des Parameters [`n.fold`] in der *rbsurv*-Funktion und der dazugehörige c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

- Iteration 2.6

  Der letzte Iterationsschritt hat wieder den Parameter [`n.seq`] im Blick. Dieser dient zur Überprüfung, ob die Bildung mehrerer Modelle eventuell einen höheren c-Index bewirkt. In Abbildung 29 wird dieser Eindruck verstärkt. So steigt der c-Index mit steigender Anzahl an Modellen. Die erhöhte Prognosegüte durch multiple Modelle hat allerdings auch eine deutlich längere Berechnungsdauer zur Folge. Abbildung 30 zeigt, dass die Berechnungsdauer von einem Modell auf drei Modelle um fast 50% zunimmt. Da es sich hierbei um die letzte Iteration handelt, wird der Parameter ungeachtet der längeren Berechnungsdauer auf [`n.seq = 3`] gesetzt.

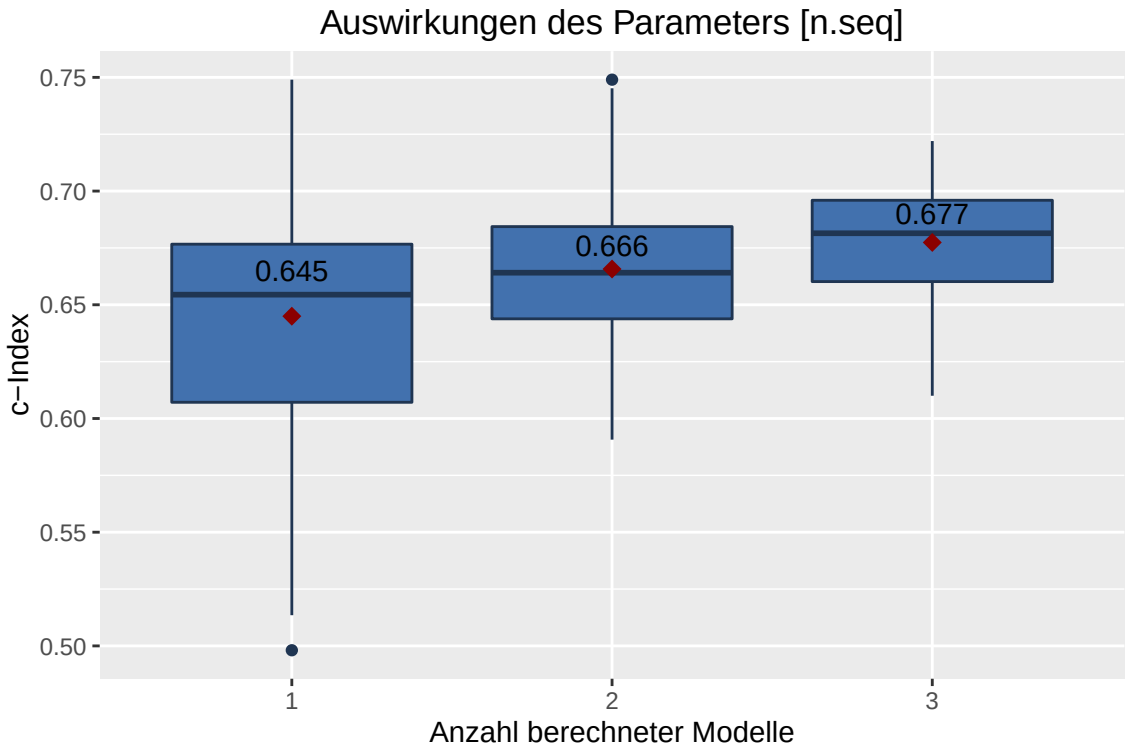

Abbildung 29: Unterschiedliche Anzahlen an multiplen Modellen und der dazugehörige c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

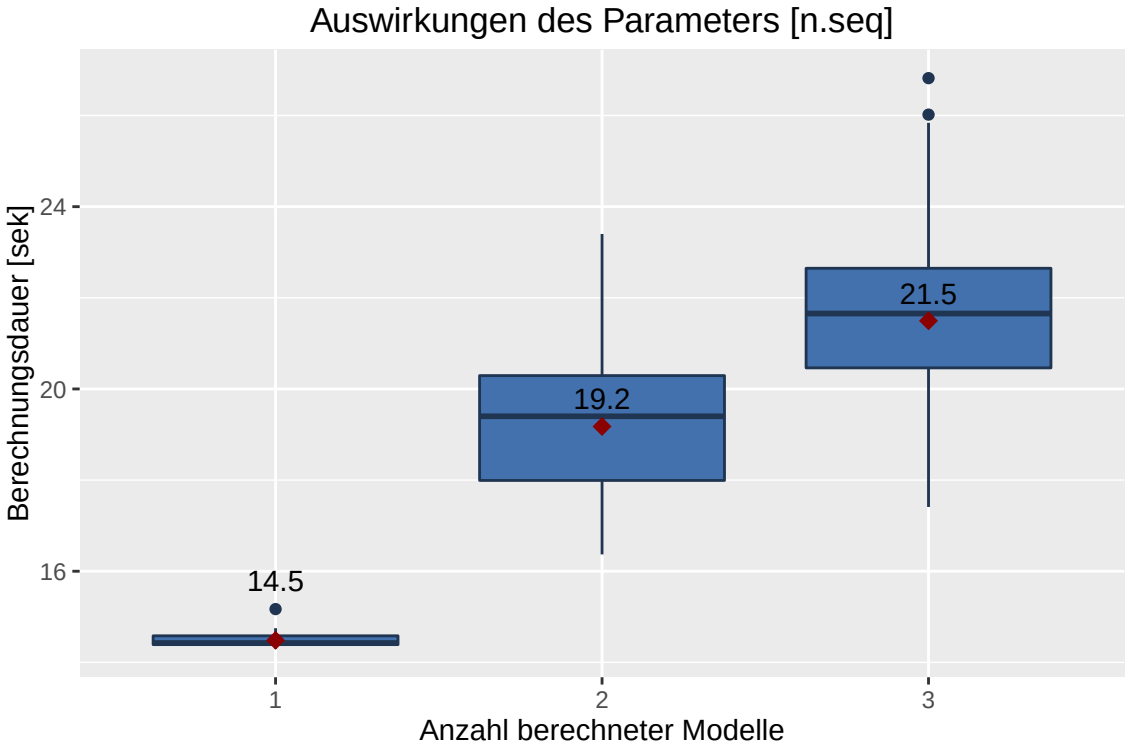

Abbildung 30: Unterschiedliche Anzahlen an multiplen Modellen und ihre Auswirkungen auf die Berechnungsdauer. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Übersicht

In Tabelle 6 befindet sich der Überblick über die einzelnen Iterationsschritte. Auch das abschließende Modell mit den optimierten Parametern befindet sich darin.

Iteration	max.n.genes	Datensatz	method	n.iter	n.fold	n.seq
Start	85	DKTK	efron	10	3	1
1.1	x	DKTK	efron	10	3	1
1.2	20	x	efron	10	3	1
1.3	20	DKTK	x	10	3	1
1.4	20	DKTK	efron	x	3	1
1.5	20	DKTK	efron	5	x	1
1.6	20	DKTK	efron	5	10	x
2.1	x	DKTK	efron	5	10	1
2.2	20	x	efron	5	10	1
2.3	20	T_80_2	x	5	10	1
2.4	20	T_80_2	efron	x	10	1
2.5	20	T_80_2	efron	5	x	1
2.6	20	T_80_2	efron	5	10	x
Ende	20	T_80_2	efron	5	10	3

Tabelle 6: Die einzelnen Simulationsschritte und die dazugehörige Parameterwahl. Die Variable x steht dafür, dass diese Variable mit mehreren Ausprägungen in die Simulation einfließt.

4.4.4 Auswirkungen der optimierten Parameterwahl

Vergleicht man das Modell mit den Default-Parametereinstellungen mit dem optimierten Modell, so lassen sich deutliche Unterschiede erkennen. Bis auf den Parameter [`method`] wurden dabei alle Parameter im Laufe der Iterationsschritte optimiert (siehe Tabelle 6). Vergleicht man den durchschnittlichen c-Index der beiden Modelle, so schneidet das optimierte Modell deutlich besser ab. Abbildung 31 zeigt die Verteilung des c-Index für je 40 Modelle pro Gruppe. Hierbei lässt sich eine Differenz des mittleren c-Index von ca. 0.15 feststellen. Auch die Varianz des c-Index mit den Default-Einstellungen ist mit ca. 0.0023 mehr als doppelt so hoch wie die Varianz des c-Index mit den optimierten Parametern (ca. 0.0011).

Betrachtet man die dazugehörige Berechnungsdauer in Abbildung 32, so lässt sich auch hier ein großer Unterschied erkennen. Die durchschnittliche Berechnungszeit der Default-Einstellung beträgt mit ca. 156 Sekunden mehr als sechs Mal so viel wie die durchschnittliche Berechnungszeit mit den optimierten Parametern (23.5 Sekunden). Dieser enorme Zeitgewinn durch die optimierten Parameter wird hauptsächlich durch die Verringerung der maximalen Anzahl an miRNA's im Modell verursacht. So konnte bereits in Abbildung 9 der große Unterschied zwischen den verschiedenen Werten für den Parameter [`max.n.genes`] ausgemacht werden.

Obwohl das optimierte Modell deutliche Verbesserungen hinsichtlich des c-Index und der Berechnungsdauer vorweist, ist keinesfalls gegeben, dass es sich dabei um die optimalen Parameter-Einstellungen handelt.

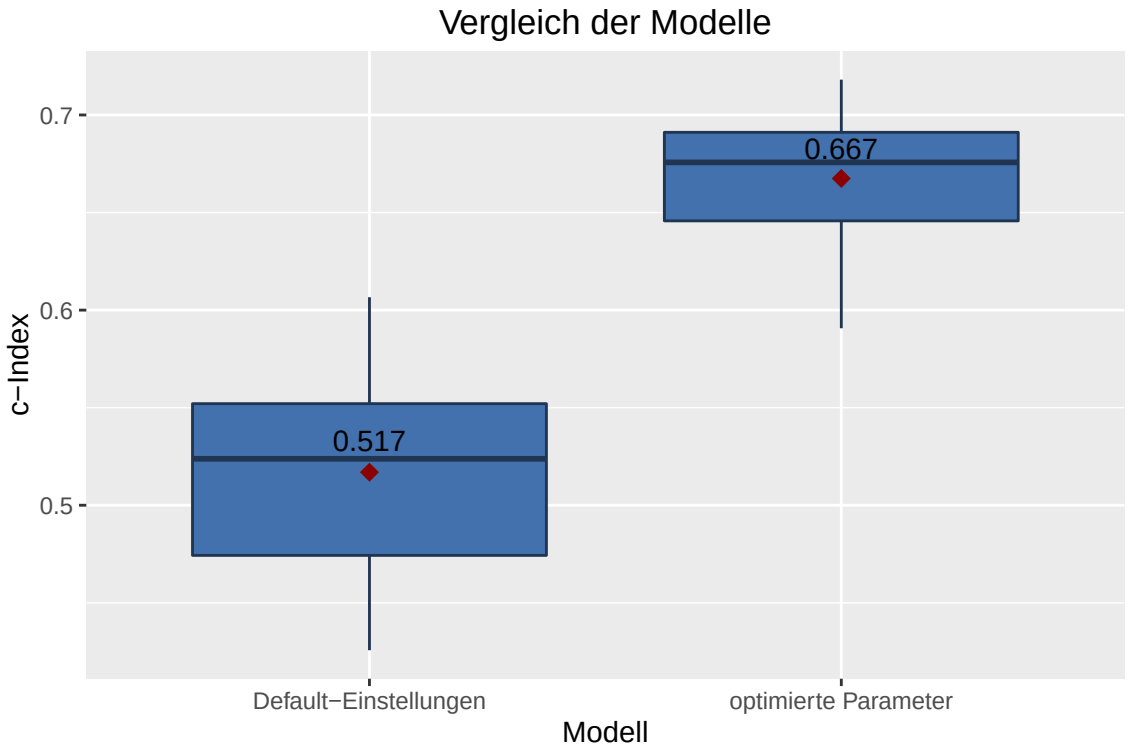

Abbildung 31: Der Vergleich zwischen den Default-Einstellungen und dem optimierten Modell bzgl. dem c-Index. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

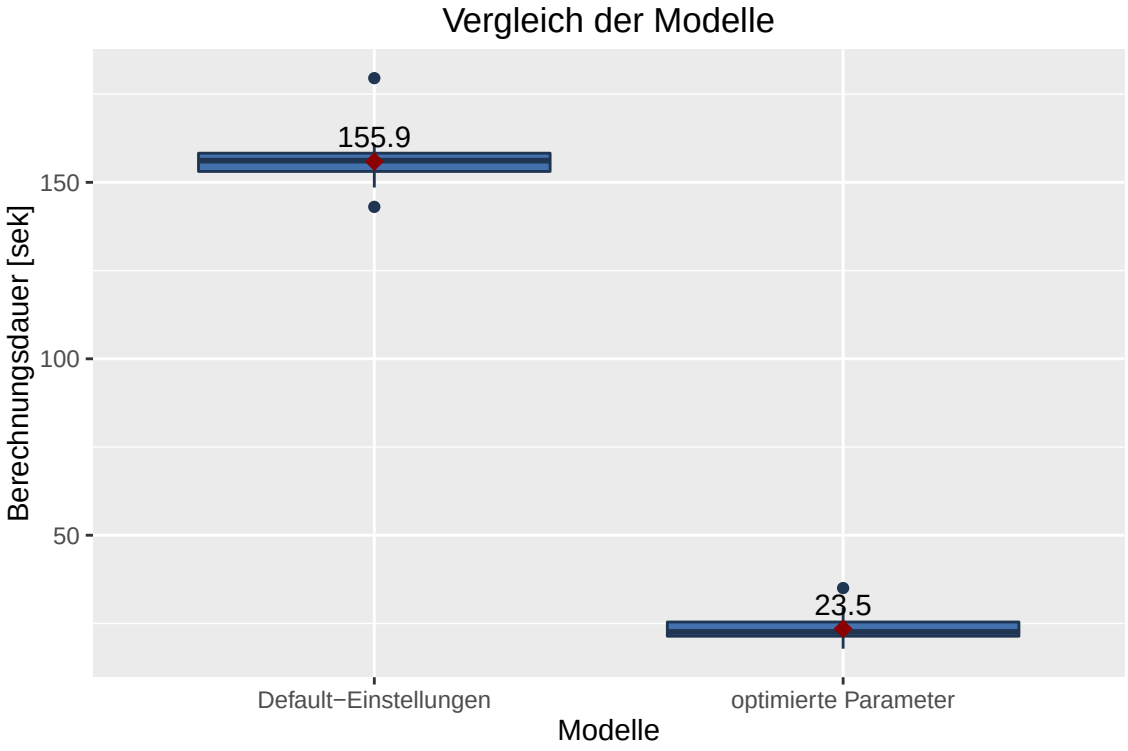

Abbildung 32: Der Vergleich zwischen den Default-Einstellungen und dem optimierten Modell bzgl. der Berechnungsdauer. Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

Betrachtet man die summierte relative Häufigkeit der 15 am häufigsten vorkommenden miRNA's im optimierten Modell, so sind die zehn häufigsten miRNA's in über 75% der Modelle enthalten. Dadurch, dass das optimierte Modell sich allerdings aus multiplen Modellen zusammensetzt, steigt die Summe der enthaltenen miRNA's automatisch an. Aus den 40 verschiedenen Modellen mit den optimierten Parametereinstellungen besitzt das beste Modell einen c-Index von ca. 0.7181 mit dem dazugehörigen Konfidenzintervall $[0.5614; 0.8749]$. Die ID-Nummern der elf enthaltenen miRNA's sind: $147, 194, 196, 443, 626, 627, 723, 801, 963, 973$ und 1000.
Das schlechteste Modell der 40 Modelle mit den optimierten Parametereinstellungen besitzt einen c-Index von ca. 0.5907 (Konfidenzintervall: $[0.3930; 0.7884]$). Dieser Unterschied liegt an den unterschiedlichen Werten für den Parameter [`seed`] und der damit verbundenen zufälligen Aufteilung in Trainings- und Validierungsdatensatz innerhalb der Kreuzvalidierung.
Interessanterweise ist das beste Modell mit den optimierten Parametereinstellungen nicht das beste Modell, welches in den Simulationsschritten zu finden war. So gab es innerhalb der Iterationsschritte Modelle, die einen höheren c-Index aufweisen können. Das beste dabei gefundene Modell ist in Iterationsschritt 2.5 berechnet worden. Im Unterschied zur optimierten Parametereinstellung wurden hier die Parameter [`n.fold = 8`] und [`n.seq = 1`] angewendet. Dadurch ergab sich ein Modell mit neun miRNA's und einem c-Index von ca. 0.7606 (Konfidenzintervall: $[0.6213; 0.8999]$).

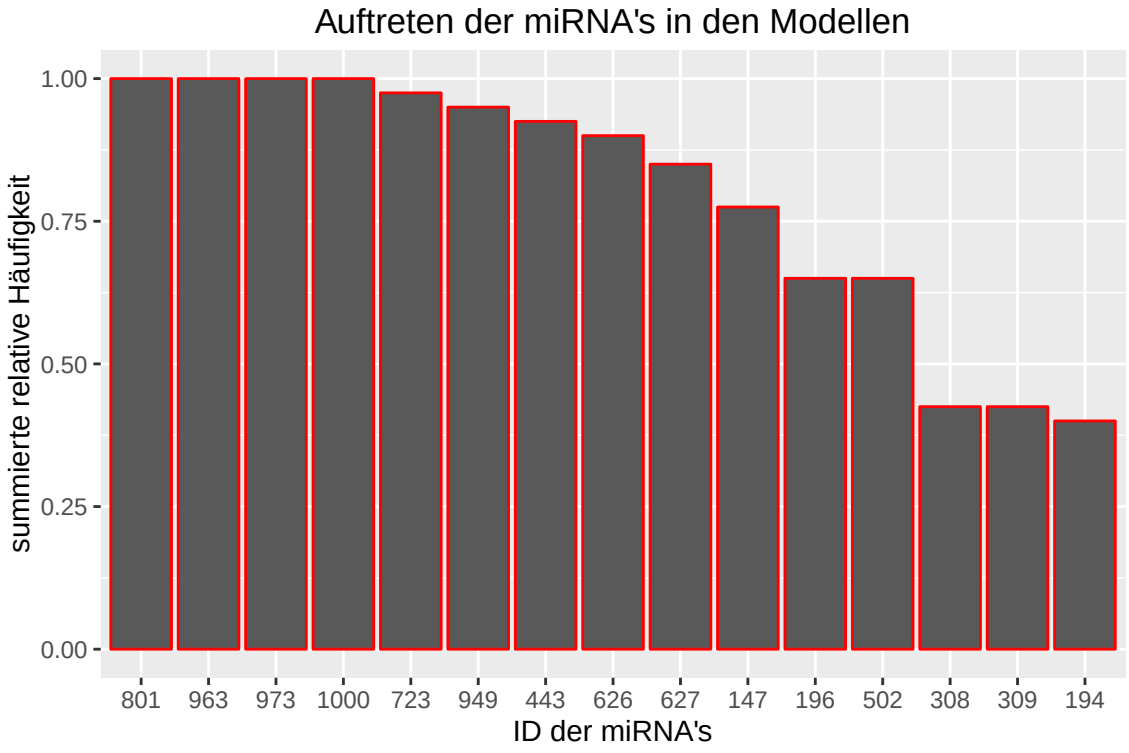

Abbildung 33: Das relative Vorkommen der 15 am häufigsten vorkommenden miRNA's in den Modellen mit optimierten Parametern. Es wurden 40 Modelle berechnet.

5 Ergebnisse

Die vorangegangene Auswertung brachte verschiedene Erkenntnisse zu Tage. Mit Hilfe von Simulationen wurden viele verschiedene Modelle mit dem R-Paket *rbsurv* berechnet und ausgewertet. Dadurch wurde die Variabilität der Funktion *rbsurv* und den darin enthaltenen Parametern aufgezeigt. Die Simulationen führten letztlich zu einem Modell mit optimierten Parametern, das deutliche Verbesserungen gegenüber dem Modell mit den Default-Parametereinstellungen aufweist. Zudem konnten Erkenntnisse für eine allgemeine optimale Parametereinstellung gewonnen werden.

5.1 Bestes Modell

Durch die zwei Simulations-Durchgänge und die darin enthaltenen Iterationsschritte wurde ein bestes Modell mit optimierter Parametereinstellung gefunden. Der Vergleich in Kapitel 4.4.4 zeigt, dass das optimierte Modell in der Prognosegüte und der Berechnungszeit deutliche Vorteile gegenüber dem Modell mit den Default-Einstellungen besitzt. Allerdings ist auch bei den optimierten Parametereinstellungen noch eine relativ große Varianz bezüglich des Parameters [`seed`] zu beobachten. Zudem gab es im Zuge der Auswertung bessere Modelle bezogen auf den c-Index. Aufgrund der Schwankungen durch den Parameter [`seed`] können somit andere Parametereinstellungen zu noch besseren Modellen führen. Die durch die Iterationen festgelegten Parameter versprechen somit nicht das beste einzelne Modell, sondern den im Durchschnitt höchsten c-Index, der hier berechneten Modelle. Durch die Verwendung von zufälligen und damit meist unterschiedlichen Werten für den Parameter [`seed`] wird der Vergleich zwischen den verschiedenen Modellen zusätzlich erschwert. Es ist dadurch nicht ersichtlich, ob der Grund für den Unterschied zwischen den Modellen die Parametereinstellung oder der Parameter [`seed`] ist.
Die in Abbildung 33 vorkommenden miRNA's sind nach dem optimierten Modell die wichtigsten miRNA's für das Wiederauftreten des Kopf-Hals-Tumors. Um ein gutes Modell zu finden empfiehlt es sich, mehrere Modelle mit den optimierten Parametern zu berechnen. Aus diesen sollte dann das beste Modell ausgesucht werden. Damit kann der Einfluss von der zufälligen Schwankung der Modellgüte durch den Parameter [`seed`] verringert werden.

5.2 Empfehlung für die Parametereinstellung

Da die erhaltenen Erkenntnisse und Ergebnisse alle auf einem einzelnen, relativ kleinen Datensatz basieren, ist es schwer daraus allgemeine Grundsätze für die Parameterwahl zu schließen. Dennoch lassen sich verschiedene Aussagen für die Parameter aus den Simulationen treffen.

1. Parameter [`seed`]

   Der Parameter [`seed`] ist der einzige Parameter, der sich nicht optimieren lässt. Dennoch zeigt z.B. Abbildung 20 die unterschiedlichen Auswirkungen, die der Parameter (abhängig von den anderen Parametern) auf die Ergebnisse hat. In Abbildung 31 ist allerdings zu sehen, dass die Varianz des c-Index durch die optimierten Parameter deutlich kleiner geworden ist. Dementsprechend liegt die Vermutung nahe, dass eine gute Einstellung der Parameter auch die Variabilität durch den Parameter [`seed`] verringert.

2. Parameter [`max.n.genes`]

   Der Parameter [`max.n.genes`] ist per Default-Einstellung auf die Anzahl der miRNA's festgelegt. Wird er nicht explizit angegeben, werden alle miRNA's auch im Algorithmus verwendet und es werden davor keine miRNA's durch univariate Modelle aussortiert. Bei der Verwendung der DKTK-Daten als Trainingsdatensatz wird die Default-Einstellung auf die Anzahl der Beobachtungen zurückgestuft, die dann 85 beträgt.
   Betrachtet man die Simulationsschritte 1.1 und 2.1, so wurde der höchste c-Index mit 10 bzw. 20 miRNA's erreicht. Ab [`max.n.genes = 30`] wurde der c-Index mit steigender Anzahl an miRNA's immer niedriger. Ebenso steigt die Berechnungsdauer der Modelle mit steigender Anzahl von miRNA's deutlich an. Es scheint sich hier tatsächlich zu lohnen, den Parameter manuell an die Daten anzupassen. Es wurde in den hier berechneten Simulationen durch eine niedrige Einstellung des Parameters [`max.n.genes`] nicht nur ein höherer c-Index, sondern auch eine deutlich geringere Berechnungsdauer erzielt.

3. Parameter [`Datensatz`]

   Als Default-Einstellung für den Parameter [`Datensatz`] wurde hier der Teildatensatz DKTK verwendet. Im Iterationsschritt 2.2 wurde diese Entscheidung zugunsten des Teildatensatzes T_80_2 verändert. Die Simulationen lassen allerdings keine Schlüsse auf eine optimale Einstellung bei multizentrischen Daten zu. So konnte weder eine Tendenz beobachtet werden, dass ein größerer Trainingsdatensatz zu besseren Ergebnissen führt, noch dass das Beibehalten der Institutsgruppen im Vergleich zu einer zufälligen Zuordnung einen positiven Unterschied ausmacht. Dementsprechend kann man allein auf der Basis dieser Daten keine Empfehlungen für den Umgang mit multizentrischen Daten aussprechen.

4. Parameter [`method`]

   Der Parameter [`method`] legt fest, nach welcher Methode Bindungen im Datensatz bei der Berechnung der Likelihood behandelt werden. Neben der Default-Einstellung von Efron (1977) gibt es noch die exakte Berechnungsmethode von Kalbfleisch und Prentice (2002) und die Methode nach Breslow (1974). Im Zuge dieser Auswertung blieb der Parameter über die gesamte Zeit bei der Default-Einstellung [`method = efron`]. Ein großer Unterschied zwischen den Methoden konnte aber nicht festgestellt werden. Der Grund hierfür ist die geringe Anzahl an Bindungen im vorhandenen Datensatz. Bei einer höheren Anzahl an Bindungen verspricht die exakte Methode nach Kalbfleisch und Prentice (2002) zwar die besten (genauesten) Ergebnisse, allerdings steigt die Berechnungszeit proportional zur Anzahl der Bindungen deutlich an. Ist die Berechnungszeit für den Anwender von hoher Bedeutung, dann ist auch bei einer hohen Zahl an Bindungen die Default-Einstellung [`method = efron`] zu empfehlen.

5. Parameter [`n.iter`]

   Betrachtet man die Simulationsschritte 1.4 und 2.4, so wurde der Parameter [`n.iter`] zweimal auf den kleinsten Wert mit [`n.iter = 5`] gesetzt. Im Vergleich zur Default-Einstellung von [`n.iter = 10`] wurde der Wert also nochmals herabgestuft. Die Vermutung, dass sich durch eine höhere Anzahl an Iterationen robustere Ergebnisse erzielen lassen können, ist durch diese Auswertung nicht zu belegen. Stattdessen nimmt vor allem in Simulationsschritt 1.4 die Variabilität des c-Index mit steigender Anzahl an Iterationen immer mehr zu. Es scheint dementsprechend oft ausreichend, wenn der Parameter [`n.iter`] relativ klein gehalten wird. Die Default-Einstellung von [`n.iter = 10`] scheint hierbei eine gute Wahl zu sein, da somit auch die Berechnungszeit in Grenzen gehalten wird.

6. Parameter [`n.fold`]

   Der Parameter [`n.fold`] dient dazu, innerhalb des Algorithmus eine Aufteilung in Trainings- und Validierungsdatensatz vorzunehmen. Diese Aufteilung erfolgt ausschließlich innerhalb des Algorithmus und hat nichts mit dem in dieser Auswertung verwendeten Validierungsdatensatz für den c-Index zu tun. Die Auswertungen ergaben dabei zweimal die Empfehlung, dass der Parameter auf [`n.fold = 10`] gesetzt werden solle. Es ließ sich mit steigenden Werten für den Parameter [`n.fold`] jeweils eine leicht positive Tendenz des c-Index beobachten. Dies legt die Vermutung nahe, dass sich ein großer interner Trainingsdatensatz positiv auf den c-Index bemerkbar macht. Da diese Einteilungen keine nennenswerten Auswirkungen auf die Berechnungszeit besitzen, ist es empfehlenswert, die Default-Einstellung [`n.fold = 3`] zu verändern. Vorsicht ist allerdings geboten, da es, je nach Datensatz, unterschiedliche Möglichkeiten zur Einstellung des Parameters gibt (siehe Formel 11).

7. Parameter [`n.seq`]

   Der Parameter [`n.seq`] dient in erster Linie dazu, voneinander unabhängige, sich ergänzende multiple Modelle zu bilden. Im Rahmen dieser Bachelorarbeit wurden bei der Bildung multipler Modelle die Einflussvariablen alle in ein Modell übernommen. Dadurch ergab sich ein höherer c-Index, allerdings nahm auch die Berechnungszeit deutlich zu. Die Einstellung dieses Parameters hängt dementsprechend stark von den Zielen des Anwenders ab. Ist ein Modell mit einer hohen Prognosegüte oder das Auffinden aller eventuell relevanter miRNA's gewünscht, so ist die Bildung multipler Modelle zu empfehlen. Das aus statistischer Sicht gesehen beste Modell (nach dem AIC) ist aber meist mit der Einstellung [`n.seq = 1`] zu erreichen. Auch die erhöhte Berechnungszeit bei der Bildung multipler Modelle kann eine Rolle für den Anwender spielen.

6 Fazit

In dieser Bachelorarbeit wurden die Auswirkungen der Parameterwahl im R-Paket *rbsurv* auf die Variablenselektion untersucht. Hierfür wurde ein multizentrischer Datensatz vom Deutschen Konsortium für translationale Krebsforschung und ein monozentrischer Datensatz der klinischen Kooperationsgruppe der LMU München und der Klinik für Strahlentherapie und Radioonkologie verwendet.

Im Zuge der Auswertung wurde ein Modell mit optimierten Parametereinstellungen gesucht. Dazu wurden zwei Durchgänge mit jeweils sechs Iterationen durchgeführt. Innerhalb einer Iteration wurde, bei Festhalten aller anderen Parameter, ein Parameter mit verschiedenen Werten geprüft und auf den besten Wert festgelegt. Als Gütekriterium wurde dabei der c-Index verwendet, der auf einem vom Modell unabhängigen Validierungsdatensatz ermittelt wurde. Als weiteres Gütekriterium wurde die Berechnungsdauer der Modelle herangezogen.
Das durch die Iterationsschritte gewonnene, optimierte Modell weißt deutliche Verbesserungen hinsichtlich des c-Index und der Berechnungsdauer auf. So konnte der durchschnittliche c-Index im Vergleich zu dem Modell mit den Default-Einstellungen deutlich erhöht werden und die mittlere Berechnungsdauer deutlich gesenkt werden. Werden die einzelnen Parameter betrachtet, so haben sie alle eine mehr oder weniger große Wirkung auf die Variablenselektion und die damit verbundene Modellgüte. Ausgenommen werden muss dabei der Parameter [`method`], der durch die geringe Anzahl an Bindungen im vorliegenden Datensatz kaum einen Einfluss hat. Eine optimale Einstellung der Parameter kann allerdings nicht pauschal angeben werden. Die Parameter und ihre optimale Einstellung hängt insbesondere auch vom verwendeten Datensatz ab und ist somit für jede Auswertung unterschiedlich. Allerdings gab es innerhalb dieser Auswertung Tendenzen zu sehen, für welche Einstellungen die Parameter die besten Ergebnisse erzielten. Da im Zuge dieser Auswertung auch verschiedene Teil-Datensätze als Trainings-Datensatz verwendet wurden, können Vermutungen angestellt werden, dass diese Tendenzen auch für andere Datensätze gelten. Um für eine neue Auswertung die richtigen Parameter-Einstellungen zu finden, ist es dennoch zu empfehlen, individuelle Simulationen zu den einzelnen Parametern durchzuführen. Trotz dieser optimierter Einstellungen ist der Parameter [`seed`] für eine gewisse Variabilität in den Ergebnissen verantwortlich. Durch eine Optimierung der Parameter konnte diese zwar verringert, aber nicht vollständig verhindert werden.

Abschließend ist festzuhalten, dass das R-Paket *rbsurv* eine gute und individuell anpassbare Möglichkeit bietet, um Micro-Array-Daten zu analysieren. Allerdings kann es durchaus aufwendig und schwierig sein die richtigen bzw. optimalen Parametereinstellungen zu finden. Die Default-Einstellungen der Funktion dienen lediglich als Ausgangspunkt und sind keineswegs für alle Datensätze auch die optimalen Einstellungen. Mit einer optimierten Parametereinstellung kann somit nicht nur die Modellgüte und die Berechnungsdauer verbessert, sondern auch die Variabilität durch den Parameter [`seed`] verringert werden.

7 Anhang

Im Anhang befindet sich weiteres Material zu dieser Bachelorarbeit. Dazu gehören sowohl weitere Abbildungen und Tabellen, wie auch die verwendeten R-Codes.

7.1 Abbildungen und Tabellen

Hier finden sich alle im vorhergehenden Text erwähnten Abbildungen und Tabellen.

ID	miRNA	ID	miRNA	ID	miRNA
1	hsa.let.7a.5p	345	hsa.miR.3692.5p	689	hsa.miR.605.5p
2	hsa.let.7b.3p	346	hsa.miR.370.3p	690	hsa.miR.6068
3	hsa.let.7b.5p	347	hsa.miR.3713	691	hsa.miR.6069
4	hsa.let.7c.5p	348	hsa.miR.371a.5p	692	hsa.miR.6073
5	hsa.let.7d.3p	349	hsa.miR.371b.5p	693	hsa.miR.6074
6	hsa.let.7d.5p	350	hsa.miR.373.5p	694	hsa.miR.6075
7	hsa.let.7e.5p	351	hsa.miR.374a.5p	695	hsa.miR.6076
8	hsa.let.7f.5p	352	hsa.miR.374b.5p	696	hsa.miR.6083
9	hsa.let.7g.3p	353	hsa.miR.376a.3p	697	hsa.miR.6084
10	hsa.let.7g.5p	354	hsa.miR.376c.3p	698	hsa.miR.6085
11	hsa.let.7i.5p	355	hsa.miR.378a.3p	699	hsa.miR.6086
12	hsa.miR.1.3p	356	hsa.miR.378b	700	hsa.miR.6087
13	hsa.miR.100.5p	357	hsa.miR.378c	701	hsa.miR.6088
14	hsa.miR.101.3p	358	hsa.miR.378d	702	hsa.miR.6089
15	hsa.miR.103a.3p	359	hsa.miR.378e	703	hsa.miR.6090
16	hsa.miR.106b.5p	360	hsa.miR.378f	704	hsa.miR.610
17	hsa.miR.107	361	hsa.miR.378g	705	hsa.miR.6124
18	hsa.miR.10a.5p	362	hsa.miR.378i	706	hsa.miR.6125
19	hsa.miR.10b.3p	363	hsa.miR.381.3p	707	hsa.miR.6126
20	hsa.miR.10b.5p	364	hsa.miR.3907	708	hsa.miR.6127
21	hsa.miR.1180.3p	365	hsa.miR.3911	709	hsa.miR.6129
22	hsa.miR.1181	366	hsa.miR.3917	710	hsa.miR.6131
23	hsa.miR.1182	367	hsa.miR.3922.5p	711	hsa.miR.6132
24	hsa.miR.1183	368	hsa.miR.3925.5p	712	hsa.miR.6133
25	hsa.miR.1185.1.3p	369	hsa.miR.3926	713	hsa.miR.6134
26	hsa.miR.1185.2.3p	370	hsa.miR.3934.3p	714	hsa.miR.614
27	hsa.miR.1199.5p	371	hsa.miR.3934.5p	715	hsa.miR.615.3p
28	hsa.miR.1202	372	hsa.miR.3935	716	hsa.miR.616.3p
29	hsa.miR.1203	373	hsa.miR.3937	717	hsa.miR.6165
30	hsa.miR.1207.5p	374	hsa.miR.3940.3p	718	hsa.miR.617
31	hsa.miR.1208	375	hsa.miR.3940.5p	719	hsa.miR.619.5p
32	hsa.miR.1224.5p	376	hsa.miR.3944.5p	720	hsa.miR.622
33	hsa.miR.1225.5p	377	hsa.miR.3945	721	hsa.miR.623
34	hsa.miR.1226.5p	378	hsa.miR.3960	722	hsa.miR.628.3p
35	hsa.miR.1227.3p	379	hsa.miR.3972	723	hsa.miR.629.3p
36	hsa.miR.1227.5p	380	hsa.miR.3976	724	hsa.miR.630
37	hsa.miR.1228.3p	381	hsa.miR.422a	725	hsa.miR.631

38	hsa.miR.1228.5p	382	hsa.miR.423.3p	726	hsa.miR.636
39	hsa.miR.1229.3p	383	hsa.miR.423.5p	727	hsa.miR.638
40	hsa.miR.1229.5p	384	hsa.miR.424.3p	728	hsa.miR.639
41	hsa.miR.1233.5p	385	hsa.miR.424.5p	729	hsa.miR.640
42	hsa.miR.1234.3p	386	hsa.miR.425.3p	730	hsa.miR.642a.3p
43	hsa.miR.1236.5p	387	hsa.miR.425.5p	731	hsa.miR.642b.3p
44	hsa.miR.1237.3p	388	hsa.miR.4251	732	hsa.miR.645
45	hsa.miR.1237.5p	389	hsa.miR.4253	733	hsa.miR.648
46	hsa.miR.1238.3p	390	hsa.miR.4257	734	hsa.miR.650
47	hsa.miR.1238.5p	391	hsa.miR.4259	735	hsa.miR.6500.3p
48	hsa.miR.1246	392	hsa.miR.4260	736	hsa.miR.6500.5p
49	hsa.miR.1247.3p	393	hsa.miR.4261	737	hsa.miR.6507.5p
50	hsa.miR.1249.3p	394	hsa.miR.4269	738	hsa.miR.6508.5p
51	hsa.miR.1249.5p	395	hsa.miR.4270	739	hsa.miR.6509.5p
52	hsa.miR.1254	396	hsa.miR.4271	740	hsa.miR.6510.5p
53	hsa.miR.125a.3p	397	hsa.miR.4274	741	hsa.miR.6511a.3p
54	hsa.miR.125a.5p	398	hsa.miR.4280	742	hsa.miR.6511a.5p
55	hsa.miR.125b.1.3p	399	hsa.miR.4281	743	hsa.miR.6511b.3p
56	hsa.miR.125b.2.3p	400	hsa.miR.4282	744	hsa.miR.6511b.5p
57	hsa.miR.125b.5p	401	hsa.miR.4284	745	hsa.miR.6512.5p
58	hsa.miR.126.3p	402	hsa.miR.4286	746	hsa.miR.6515.3p
59	hsa.miR.126.5p	403	hsa.miR.429	747	hsa.miR.6516.3p
60	hsa.miR.1260a	404	hsa.miR.4291	748	hsa.miR.6516.5p
61	hsa.miR.1260b	405	hsa.miR.4294	749	hsa.miR.652.5p
62	hsa.miR.1261	406	hsa.miR.4298	750	hsa.miR.654.5p
63	hsa.miR.1266.3p	407	hsa.miR.4299	751	hsa.miR.658
64	hsa.miR.1268a	408	hsa.miR.4300	752	hsa.miR.659.3p
65	hsa.miR.1268b	409	hsa.miR.4304	753	hsa.miR.660.5p
66	hsa.miR.127.3p	410	hsa.miR.4306	754	hsa.miR.662
67	hsa.miR.1273c	411	hsa.miR.431.5p	755	hsa.miR.663a
68	hsa.miR.1273d	412	hsa.miR.4311	756	hsa.miR.663b
69	hsa.miR.1273e	413	hsa.miR.4312	757	hsa.miR.664a.3p
70	hsa.miR.1273f	414	hsa.miR.4313	758	hsa.miR.664a.5p
71	hsa.miR.1273g.3p	415	hsa.miR.4314	759	hsa.miR.664b.3p
72	hsa.miR.1273g.5p	416	hsa.miR.4317	760	hsa.miR.664b.5p
73	hsa.miR.1273h.3p	417	hsa.miR.432.5p	761	hsa.miR.665
74	hsa.miR.1273h.5p	418	hsa.miR.4322	762	hsa.miR.668.3p
75	hsa.miR.1275	419	hsa.miR.4323	763	hsa.miR.671.3p
76	hsa.miR.1276	420	hsa.miR.4324	764	hsa.miR.671.5p
77	hsa.miR.128.1.5p	421	hsa.miR.4327	765	hsa.miR.6716.5p
78	hsa.miR.128.3p	422	hsa.miR.4417	766	hsa.miR.6717.5p
79	hsa.miR.1281	423	hsa.miR.4418	767	hsa.miR.6718.5p
80	hsa.miR.1285.3p	424	hsa.miR.4419a	768	hsa.miR.6720.3p
81	hsa.miR.1285.5p	425	hsa.miR.4419b	769	hsa.miR.6720.5p
82	hsa.miR.1287.5p	426	hsa.miR.4421	770	hsa.miR.6722.3p

83	hsa.miR.1288.3p	427	hsa.miR.4425	771	hsa.miR.6722.5p
84	hsa.miR.129.5p	428	hsa.miR.4428	772	hsa.miR.6723.5p
85	hsa.miR.1290	429	hsa.miR.4429	773	hsa.miR.6724.5p
86	hsa.miR.1291	430	hsa.miR.4430	774	hsa.miR.6726.5p
87	hsa.miR.1295a	431	hsa.miR.4433a.3p	775	hsa.miR.6727.5p
88	hsa.miR.1295b.3p	432	hsa.miR.4433a.5p	776	hsa.miR.6728.5p
89	hsa.miR.1296.5p	433	hsa.miR.4433b.3p	777	hsa.miR.6729.5p
90	hsa.miR.1299	434	hsa.miR.4436a	778	hsa.miR.6730.3p
91	hsa.miR.1301.5p	435	hsa.miR.4436b.3p	779	hsa.miR.6730.5p
92	hsa.miR.1303	436	hsa.miR.4436b.5p	780	hsa.miR.6731.3p
93	hsa.miR.1304.3p	437	hsa.miR.4441	781	hsa.miR.6732.3p
94	hsa.miR.1305	438	hsa.miR.4442	782	hsa.miR.6732.5p
95	hsa.miR.1306.3p	439	hsa.miR.4443	783	hsa.miR.6734.5p
96	hsa.miR.1307.3p	440	hsa.miR.4444	784	hsa.miR.6736.5p
97	hsa.miR.1307.5p	441	hsa.miR.4446.3p	785	hsa.miR.6737.3p
98	hsa.miR.130a.3p	442	hsa.miR.4447	786	hsa.miR.6737.5p
99	hsa.miR.130b.3p	443	hsa.miR.4448	787	hsa.miR.6738.5p
100	hsa.miR.132.3p	444	hsa.miR.4449	788	hsa.miR.6739.5p
101	hsa.miR.1321	445	hsa.miR.4450	789	hsa.miR.6740.5p
102	hsa.miR.1323	446	hsa.miR.4451	790	hsa.miR.6741.5p
103	hsa.miR.133a.3p	447	hsa.miR.4453	791	hsa.miR.6743.3p
104	hsa.miR.133b	448	hsa.miR.4455	792	hsa.miR.6743.5p
105	hsa.miR.134.5p	449	hsa.miR.4458	793	hsa.miR.6745
106	hsa.miR.1343.5p	450	hsa.miR.4459	794	hsa.miR.6746.5p
107	hsa.miR.135a.3p	451	hsa.miR.4462	795	hsa.miR.6747.5p
108	hsa.miR.135b.5p	452	hsa.miR.4463	796	hsa.miR.6748.5p
109	hsa.miR.138.2.3p	453	hsa.miR.4465	797	hsa.miR.6749.5p
110	hsa.miR.139.3p	454	hsa.miR.4466	798	hsa.miR.6751.3p
111	hsa.miR.140.3p	455	hsa.miR.4468	799	hsa.miR.6752.3p
112	hsa.miR.140.5p	456	hsa.miR.4470	800	hsa.miR.6752.5p
113	hsa.miR.141.3p	457	hsa.miR.4472	801	hsa.miR.6753.3p
114	hsa.miR.142.3p	458	hsa.miR.4475	802	hsa.miR.6753.5p
115	hsa.miR.142.5p	459	hsa.miR.4476	803	hsa.miR.6754.5p
116	hsa.miR.143.3p	460	hsa.miR.4478	804	hsa.miR.6756.3p
117	hsa.miR.144.3p	461	hsa.miR.4481	805	hsa.miR.6756.5p
118	hsa.miR.145.5p	462	hsa.miR.4482.3p	806	hsa.miR.6757.5p
119	hsa.miR.1469	463	hsa.miR.4484	807	hsa.miR.6758.5p
120	hsa.miR.146a.5p	464	hsa.miR.4485.3p	808	hsa.miR.6759.3p
121	hsa.miR.146b.5p	465	hsa.miR.4485.5p	809	hsa.miR.6760.3p
122	hsa.miR.1470	466	hsa.miR.4486	810	hsa.miR.6760.5p
123	hsa.miR.1471	467	hsa.miR.4487	811	hsa.miR.6762.5p
124	hsa.miR.148a.3p	468	hsa.miR.4488	812	hsa.miR.6763.3p
125	hsa.miR.148b.3p	469	hsa.miR.4489	813	hsa.miR.6763.5p
126	hsa.miR.149.3p	470	hsa.miR.4494	814	hsa.miR.6765.3p
127	hsa.miR.149.5p	471	hsa.miR.4496	815	hsa.miR.6765.5p

128	hsa.miR.150.3p	472	hsa.miR.4497	816	hsa.miR.6766.3p
129	hsa.miR.150.5p	473	hsa.miR.4498	817	hsa.miR.6767.5p
130	hsa.miR.151a.3p	474	hsa.miR.4499	818	hsa.miR.6768.5p
131	hsa.miR.151a.5p	475	hsa.miR.449b.3p	819	hsa.miR.6769a.5p
132	hsa.miR.151b	476	hsa.miR.4502	820	hsa.miR.6769b.5p
133	hsa.miR.152.3p	477	hsa.miR.4505	821	hsa.miR.6771.5p
134	hsa.miR.1539	478	hsa.miR.4506	822	hsa.miR.6772.5p
135	hsa.miR.155.5p	479	hsa.miR.4507	823	hsa.miR.6774.5p
136	hsa.miR.1587	480	hsa.miR.4508	824	hsa.miR.6775.3p
137	hsa.miR.15a.5p	481	hsa.miR.4510	825	hsa.miR.6775.5p
138	hsa.miR.15b.5p	482	hsa.miR.4513	826	hsa.miR.6776.5p
139	hsa.miR.16.5p	483	hsa.miR.4514	827	hsa.miR.6777.3p
140	hsa.miR.17.3p	484	hsa.miR.4515	828	hsa.miR.6777.5p
141	hsa.miR.17.5p	485	hsa.miR.4516	829	hsa.miR.6778.5p
142	hsa.miR.181a.5p	486	hsa.miR.4518	830	hsa.miR.6779.3p
143	hsa.miR.181b.5p	487	hsa.miR.4519	831	hsa.miR.6779.5p
144	hsa.miR.181d.5p	488	hsa.miR.451a	832	hsa.miR.6780a.5p
145	hsa.miR.1825	489	hsa.miR.452.5p	833	hsa.miR.6780b.5p
146	hsa.miR.183.3p	490	hsa.miR.4522	834	hsa.miR.6781.5p
147	hsa.miR.184	491	hsa.miR.4526	835	hsa.miR.6782.5p
148	hsa.miR.185.5p	492	hsa.miR.4530	836	hsa.miR.6784.3p
149	hsa.miR.186.5p	493	hsa.miR.4531	837	hsa.miR.6784.5p
150	hsa.miR.187.5p	494	hsa.miR.4532	838	hsa.miR.6785.3p
151	hsa.miR.188.5p	495	hsa.miR.4533	839	hsa.miR.6785.5p
152	hsa.miR.18a.5p	496	hsa.miR.4534	840	hsa.miR.6786.5p
153	hsa.miR.1908.3p	497	hsa.miR.4535	841	hsa.miR.6787.3p
154	hsa.miR.1909.5p	498	hsa.miR.4538	842	hsa.miR.6787.5p
155	hsa.miR.191.3p	499	hsa.miR.4539	843	hsa.miR.6788.5p
156	hsa.miR.1910.3p	500	hsa.miR.455.3p	844	hsa.miR.6789.5p
157	hsa.miR.1910.5p	501	hsa.miR.4632.5p	845	hsa.miR.6790.3p
158	hsa.miR.1913	502	hsa.miR.4633.5p	846	hsa.miR.6790.5p
159	hsa.miR.1914.3p	503	hsa.miR.4634	847	hsa.miR.6791.5p
160	hsa.miR.1915.3p	504	hsa.miR.4636	848	hsa.miR.6792.3p
161	hsa.miR.193a.3p	505	hsa.miR.4640.5p	849	hsa.miR.6792.5p
162	hsa.miR.193a.5p	506	hsa.miR.4642	850	hsa.miR.6793.5p
163	hsa.miR.193b.3p	507	hsa.miR.4644	851	hsa.miR.6794.5p
164	hsa.miR.193b.5p	508	hsa.miR.4646.5p	852	hsa.miR.6795.5p
165	hsa.miR.194.3p	509	hsa.miR.4647	853	hsa.miR.6796.3p
166	hsa.miR.195.3p	510	hsa.miR.4648	854	hsa.miR.6796.5p
167	hsa.miR.195.5p	511	hsa.miR.4649.3p	855	hsa.miR.6797.3p
168	hsa.miR.196a.5p	512	hsa.miR.4651	856	hsa.miR.6797.5p
169	hsa.miR.196b.5p	513	hsa.miR.4653.3p	857	hsa.miR.6798.3p
170	hsa.miR.197.3p	514	hsa.miR.4654	858	hsa.miR.6798.5p
171	hsa.miR.197.5p	515	hsa.miR.4655.3p	859	hsa.miR.6799.5p
172	hsa.miR.1972	516	hsa.miR.4655.5p	860	hsa.miR.6800.3p

173	hsa.miR.1973	517	hsa.miR.4656	861	hsa.miR.6800.5p
174	hsa.miR.198	518	hsa.miR.4657	862	hsa.miR.6801.3p
175	hsa.miR.199a.3p	519	hsa.miR.4659a.3p	863	hsa.miR.6801.5p
176	hsa.miR.199a.5p	520	hsa.miR.4659b.3p	864	hsa.miR.6802.5p
177	hsa.miR.199b.5p	521	hsa.miR.4660	865	hsa.miR.6803.5p
178	hsa.miR.19a.3p	522	hsa.miR.4664.3p	866	hsa.miR.6804.3p
179	hsa.miR.19b.3p	523	hsa.miR.4665.3p	867	hsa.miR.6804.5p
180	hsa.miR.200a.3p	524	hsa.miR.4665.5p	868	hsa.miR.6805.5p
181	hsa.miR.200a.5p	525	hsa.miR.4667.5p	869	hsa.miR.6806.5p
182	hsa.miR.200b.3p	526	hsa.miR.4668.5p	870	hsa.miR.6807.5p
183	hsa.miR.200b.5p	527	hsa.miR.4669	871	hsa.miR.6808.5p
184	hsa.miR.200c.3p	528	hsa.miR.4672	872	hsa.miR.6809.5p
185	hsa.miR.202.3p	529	hsa.miR.4673	873	hsa.miR.6812.3p
186	hsa.miR.203a.3p	530	hsa.miR.4674	874	hsa.miR.6812.5p
187	hsa.miR.204.5p	531	hsa.miR.4675	875	hsa.miR.6813.3p
188	hsa.miR.205.3p	532	hsa.miR.4676.5p	876	hsa.miR.6815.5p
189	hsa.miR.205.5p	533	hsa.miR.4682	877	hsa.miR.6817.5p
190	hsa.miR.206	534	hsa.miR.4685.5p	878	hsa.miR.6819.3p
191	hsa.miR.208a.5p	535	hsa.miR.4687.3p	879	hsa.miR.6819.5p
192	hsa.miR.20a.5p	536	hsa.miR.4688	880	hsa.miR.6820.3p
193	hsa.miR.20b.5p	537	hsa.miR.4689	881	hsa.miR.6820.5p
194	hsa.miR.21.3p	538	hsa.miR.4690.5p	882	hsa.miR.6821.5p
195	hsa.miR.21.5p	539	hsa.miR.4691.5p	883	hsa.miR.6824.3p
196	hsa.miR.210.3p	540	hsa.miR.4695.3p	884	hsa.miR.6824.5p
197	hsa.miR.210.5p	541	hsa.miR.4695.5p	885	hsa.miR.6825.3p
198	hsa.miR.211.3p	542	hsa.miR.4697.5p	886	hsa.miR.6825.5p
199	hsa.miR.2117	543	hsa.miR.4698	887	hsa.miR.6826.5p
200	hsa.miR.212.3p	544	hsa.miR.4701.3p	888	hsa.miR.6829.5p
201	hsa.miR.214.3p	545	hsa.miR.4701.5p	889	hsa.miR.6830.5p
202	hsa.miR.22.3p	546	hsa.miR.4706	890	hsa.miR.6831.5p
203	hsa.miR.221.3p	547	hsa.miR.4707.3p	891	hsa.miR.6833.5p
204	hsa.miR.222.3p	548	hsa.miR.4707.5p	892	hsa.miR.6834.3p
205	hsa.miR.223.3p	549	hsa.miR.4709.3p	893	hsa.miR.6836.3p
206	hsa.miR.224.3p	550	hsa.miR.4710	894	hsa.miR.6837.5p
207	hsa.miR.224.5p	551	hsa.miR.4713.3p	895	hsa.miR.6839.5p
208	hsa.miR.2276.3p	552	hsa.miR.4715.5p	896	hsa.miR.6840.3p
209	hsa.miR.2277.3p	553	hsa.miR.4716.3p	897	hsa.miR.6845.5p
210	hsa.miR.2392	554	hsa.miR.4717.3p	898	hsa.miR.6846.5p
211	hsa.miR.23a.3p	555	hsa.miR.4721	899	hsa.miR.6847.5p
212	hsa.miR.23a.5p	556	hsa.miR.4725.3p	900	hsa.miR.6848.3p
213	hsa.miR.23b.3p	557	hsa.miR.4725.5p	901	hsa.miR.6848.5p
214	hsa.miR.24.3p	558	hsa.miR.4726.5p	902	hsa.miR.6849.5p
215	hsa.miR.2467.3p	559	hsa.miR.4728.3p	903	hsa.miR.6850.5p
216	hsa.miR.25.3p	560	hsa.miR.4728.5p	904	hsa.miR.6851.3p
217	hsa.miR.26a.5p	561	hsa.miR.4731.3p	905	hsa.miR.6851.5p

218	hsa.miR.26b.5p	562	hsa.miR.4732.5p	906	hsa.miR.6855.5p
219	hsa.miR.27a.3p	563	hsa.miR.4733.5p	907	hsa.miR.6856.5p
220	hsa.miR.27b.3p	564	hsa.miR.4734	908	hsa.miR.6857.5p
221	hsa.miR.28.3p	565	hsa.miR.4738.3p	909	hsa.miR.6858.3p
222	hsa.miR.28.5p	566	hsa.miR.4739	910	hsa.miR.6858.5p
223	hsa.miR.2861	567	hsa.miR.4740.5p	911	hsa.miR.6859.5p
224	hsa.miR.296.5p	568	hsa.miR.4741	912	hsa.miR.6860
225	hsa.miR.298	569	hsa.miR.4743.5p	913	hsa.miR.6861.3p
226	hsa.miR.29a.3p	570	hsa.miR.4745.5p	914	hsa.miR.6861.5p
227	hsa.miR.29b.3p	571	hsa.miR.4746.3p	915	hsa.miR.6862.5p
228	hsa.miR.29c.3p	572	hsa.miR.4746.5p	916	hsa.miR.6865.3p
229	hsa.miR.29c.5p	573	hsa.miR.4748	917	hsa.miR.6865.5p
230	hsa.miR.301a.3p	574	hsa.miR.4749.3p	918	hsa.miR.6867.5p
231	hsa.miR.302c.5p	575	hsa.miR.4749.5p	919	hsa.miR.6869.5p
232	hsa.miR.30a.5p	576	hsa.miR.4750.3p	920	hsa.miR.6870.3p
233	hsa.miR.30b.3p	577	hsa.miR.4750.5p	921	hsa.miR.6870.5p
234	hsa.miR.30b.5p	578	hsa.miR.4753.5p	922	hsa.miR.6871.5p
235	hsa.miR.30c.1.3p	579	hsa.miR.4755.3p	923	hsa.miR.6872.3p
236	hsa.miR.30c.2.3p	580	hsa.miR.4758.3p	924	hsa.miR.6872.5p
237	hsa.miR.30c.5p	581	hsa.miR.4758.5p	925	hsa.miR.6873.5p
238	hsa.miR.30d.5p	582	hsa.miR.4763.3p	926	hsa.miR.6875.3p
239	hsa.miR.30e.3p	583	hsa.miR.4763.5p	927	hsa.miR.6875.5p
240	hsa.miR.30e.5p	584	hsa.miR.4767	928	hsa.miR.6876.5p
241	hsa.miR.31.3p	585	hsa.miR.4768.3p	929	hsa.miR.6877.3p
242	hsa.miR.31.5p	586	hsa.miR.4769.3p	930	hsa.miR.6877.5p
243	hsa.miR.3121.3p	587	hsa.miR.4769.5p	931	hsa.miR.6879.5p
244	hsa.miR.3122	588	hsa.miR.4773	932	hsa.miR.6880.3p
245	hsa.miR.3124.5p	589	hsa.miR.4776.5p	933	hsa.miR.6880.5p
246	hsa.miR.3125	590	hsa.miR.4778.5p	934	hsa.miR.6881.5p
247	hsa.miR.3127.5p	591	hsa.miR.4783.3p	935	hsa.miR.6882.5p
248	hsa.miR.3130.5p	592	hsa.miR.4784	936	hsa.miR.6885.3p
249	hsa.miR.3131	593	hsa.miR.4785	937	hsa.miR.6886.3p
250	hsa.miR.3132	594	hsa.miR.4787.3p	938	hsa.miR.6886.5p
251	hsa.miR.3135b	595	hsa.miR.4787.5p	939	hsa.miR.6887.5p
252	hsa.miR.3137	596	hsa.miR.4788	940	hsa.miR.6889.3p
253	hsa.miR.3138	597	hsa.miR.4792	941	hsa.miR.6889.5p
254	hsa.miR.3141	598	hsa.miR.4793.3p	942	hsa.miR.6890.3p
255	hsa.miR.3147	599	hsa.miR.4793.5p	943	hsa.miR.6890.5p
256	hsa.miR.3150b.3p	600	hsa.miR.4800.3p	944	hsa.miR.6891.5p
257	hsa.miR.3150b.5p	601	hsa.miR.4800.5p	945	hsa.miR.6892.5p
258	hsa.miR.3151.3p	602	hsa.miR.483.3p	946	hsa.miR.6893.5p
259	hsa.miR.3154	603	hsa.miR.483.5p	947	hsa.miR.6894.5p
260	hsa.miR.3155b	604	hsa.miR.484	948	hsa.miR.6895.5p
261	hsa.miR.3156.5p	605	hsa.miR.486.5p	949	hsa.miR.7.5p
262	hsa.miR.3158.5p	606	hsa.miR.487b.3p	950	hsa.miR.708.5p

263	hsa.miR.3161	607	hsa.miR.487b.5p	951	hsa.miR.7106.5p
264	hsa.miR.3162.3p	608	hsa.miR.489.3p	952	hsa.miR.7107.5p
265	hsa.miR.3162.5p	609	hsa.miR.490.5p	953	hsa.miR.7108.3p
266	hsa.miR.3163	610	hsa.miR.492	954	hsa.miR.7108.5p
267	hsa.miR.3173.3p	611	hsa.miR.493.3p	955	hsa.miR.7109.3p
268	hsa.miR.3174	612	hsa.miR.494.3p	956	hsa.miR.7109.5p
269	hsa.miR.3176	613	hsa.miR.497.5p	957	hsa.miR.711
270	hsa.miR.3177.3p	614	hsa.miR.498	958	hsa.miR.7110.5p
271	hsa.miR.3180.3p	615	hsa.miR.5001.5p	959	hsa.miR.7111.3p
272	hsa.miR.3180.5p	616	hsa.miR.5003.3p	960	hsa.miR.7111.5p
273	hsa.miR.3185	617	hsa.miR.5003.5p	961	hsa.miR.7113.5p
274	hsa.miR.3187.3p	618	hsa.miR.5006.5p	962	hsa.miR.7114.3p
275	hsa.miR.3188	619	hsa.miR.5008.5p	963	hsa.miR.7114.5p
276	hsa.miR.3189.3p	620	hsa.miR.500a.3p	964	hsa.miR.7150
277	hsa.miR.3189.5p	621	hsa.miR.500a.5p	965	hsa.miR.7151.3p
278	hsa.miR.3190.3p	622	hsa.miR.501.3p	966	hsa.miR.7152.3p
279	hsa.miR.3194.5p	623	hsa.miR.501.5p	967	hsa.miR.7152.5p
280	hsa.miR.3195	624	hsa.miR.5010.5p	968	hsa.miR.7155.3p
281	hsa.miR.3196	625	hsa.miR.504.3p	969	hsa.miR.7155.5p
282	hsa.miR.3197	626	hsa.miR.508.5p	970	hsa.miR.7156.3p
283	hsa.miR.3198	627	hsa.miR.5088.5p	971	hsa.miR.7157.5p
284	hsa.miR.3200.5p	628	hsa.miR.509.3.5p	972	hsa.miR.7159.5p
285	hsa.miR.3202	629	hsa.miR.509.5p	973	hsa.miR.7161.3p
286	hsa.miR.320a	630	hsa.miR.5090	974	hsa.miR.7162.3p
287	hsa.miR.320b	631	hsa.miR.5093	975	hsa.miR.718
288	hsa.miR.320c	632	hsa.miR.5096	976	hsa.miR.7515
289	hsa.miR.320d	633	hsa.miR.5100	977	hsa.miR.758.5p
290	hsa.miR.320e	634	hsa.miR.512.3p	978	hsa.miR.760
291	hsa.miR.324.3p	635	hsa.miR.513a.5p	979	hsa.miR.762
292	hsa.miR.324.5p	636	hsa.miR.513b.5p	980	hsa.miR.7641
293	hsa.miR.328.3p	637	hsa.miR.513c.3p	981	hsa.miR.765
294	hsa.miR.328.5p	638	hsa.miR.513c.5p	982	hsa.miR.766.3p
295	hsa.miR.330.3p	639	hsa.miR.514b.5p	983	hsa.miR.769.3p
296	hsa.miR.331.3p	640	hsa.miR.516a.5p	984	hsa.miR.770.5p
297	hsa.miR.338.5p	641	hsa.miR.516b.5p	985	hsa.miR.7704
298	hsa.miR.339.3p	642	hsa.miR.5187.5p	986	hsa.miR.7845.5p
299	hsa.miR.33b.3p	643	hsa.miR.5189.5p	987	hsa.miR.7846.3p
300	hsa.miR.342.3p	644	hsa.miR.518a.5p	988	hsa.miR.7847.3p
301	hsa.miR.345.3p	645	hsa.miR.5190	989	hsa.miR.7851.3p
302	hsa.miR.345.5p	646	hsa.miR.5194	990	hsa.miR.7854.3p
303	hsa.miR.34a.5p	647	hsa.miR.5195.3p	991	hsa.miR.7974
304	hsa.miR.34b.5p	648	hsa.miR.5195.5p	992	hsa.miR.7975
305	hsa.miR.34c.3p	649	hsa.miR.5196.5p	993	hsa.miR.7977
306	hsa.miR.3605.5p	650	hsa.miR.519e.5p	994	hsa.miR.8052
307	hsa.miR.3607.3p	651	hsa.miR.520b	995	hsa.miR.8055

308	hsa.miR.3607.5p	652	hsa.miR.520e	996	hsa.miR.8060
309	hsa.miR.3609	653	hsa.miR.526b.5p	997	hsa.miR.8063
310	hsa.miR.361.3p	654	hsa.miR.532.5p	998	hsa.miR.8064
311	hsa.miR.361.5p	655	hsa.miR.542.5p	999	hsa.miR.8069
312	hsa.miR.3610	656	hsa.miR.548q	1000	hsa.miR.8071
313	hsa.miR.3614.5p	657	hsa.miR.550a.3.5p	1001	hsa.miR.8072
314	hsa.miR.3617.3p	658	hsa.miR.550a.5p	1002	hsa.miR.8073
315	hsa.miR.3617.5p	659	hsa.miR.550b.2.5p	1003	hsa.miR.8075
316	hsa.miR.3620.3p	660	hsa.miR.551b.5p	1004	hsa.miR.8078
317	hsa.miR.3620.5p	661	hsa.miR.557	1005	hsa.miR.8085
318	hsa.miR.3621	662	hsa.miR.5572	1006	hsa.miR.8087
319	hsa.miR.3622a.5p	663	hsa.miR.5580.3p	1007	hsa.miR.8088
320	hsa.miR.3622b.3p	664	hsa.miR.5581.5p	1008	hsa.miR.8089
321	hsa.miR.3622b.5p	665	hsa.miR.5585.3p	1009	hsa.miR.8485
322	hsa.miR.3646	666	hsa.miR.5587.5p	1010	hsa.miR.874.3p
323	hsa.miR.3648	667	hsa.miR.564	1011	hsa.miR.877.5p
324	hsa.miR.3651	668	hsa.miR.566	1012	hsa.miR.885.5p
325	hsa.miR.3652	669	hsa.miR.5684	1013	hsa.miR.887.3p
326	hsa.miR.3653.3p	670	hsa.miR.5685	1014	hsa.miR.892b
327	hsa.miR.3654	671	hsa.miR.5696	1015	hsa.miR.921
328	hsa.miR.3656	672	hsa.miR.5699.5p	1016	hsa.miR.92a.3p
329	hsa.miR.3659	673	hsa.miR.5703	1017	hsa.miR.92b.3p
330	hsa.miR.365a.3p	674	hsa.miR.5708	1018	hsa.miR.93.5p
331	hsa.miR.365a.5p	675	hsa.miR.572	1019	hsa.miR.933
332	hsa.miR.365b.5p	676	hsa.miR.5739	1020	hsa.miR.934
333	hsa.miR.3660	677	hsa.miR.574.3p	1021	hsa.miR.936
334	hsa.miR.3663.3p	678	hsa.miR.574.5p	1022	hsa.miR.937.5p
335	hsa.miR.3663.5p	679	hsa.miR.575	1023	hsa.miR.939.3p
336	hsa.miR.3665	680	hsa.miR.5787	1024	hsa.miR.939.5p
337	hsa.miR.3666	681	hsa.miR.583	1025	hsa.miR.940
338	hsa.miR.3667.5p	682	hsa.miR.584.5p	1026	hsa.miR.9500
339	hsa.miR.3678.3p	683	hsa.miR.590.5p	1027	hsa.miR.96.5p
340	hsa.miR.3679.5p	684	hsa.miR.595	1028	hsa.miR.98.5p
341	hsa.miR.3680.3p	685	hsa.miR.596	1029	hsa.miR.99a.5p
342	hsa.miR.3682.3p	686	hsa.miR.598.5p	1030	hsa.miR.99b.3p
343	hsa.miR.3689a.5p	687	hsa.miR.601	1031	hsa.miR.99b.5p
344	hsa.miR.3689f	688	hsa.miR.602		

Tabelle 7: Die einzelnen miRNA's zusammen mit ihren ID-Nummern

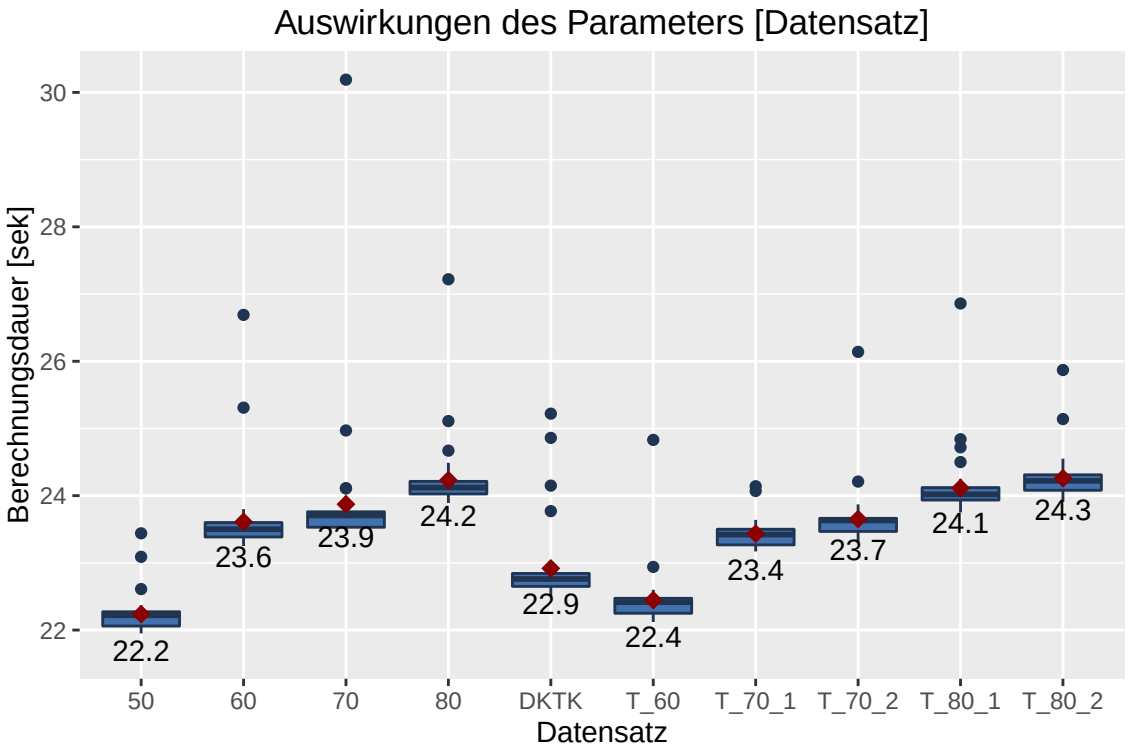

Abbildung 34: Die verschiedenen Trainings-Datensätze und ihre dazugehörige Berechnungsdauer (Iter.: 1.2). Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

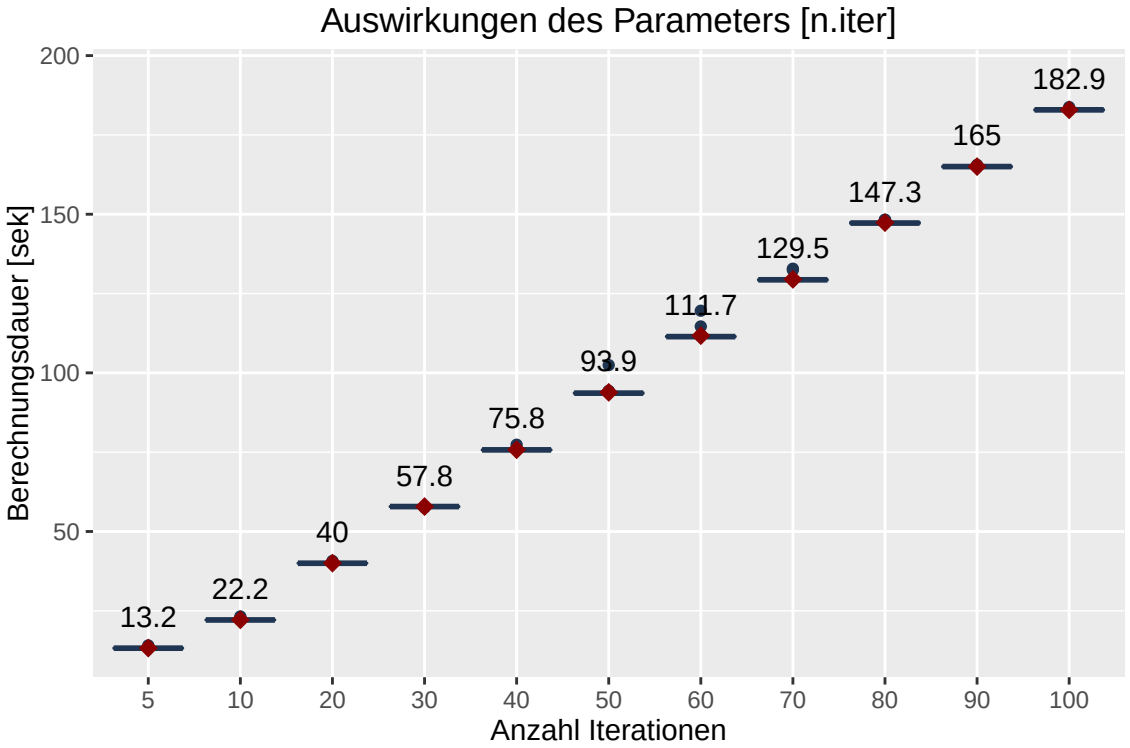

Abbildung 35: Die verschiedenen Werte für die Anzahl an Iterationen und ihre dazugehörige Berechnungsdauer (Iter.: 1.4). Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

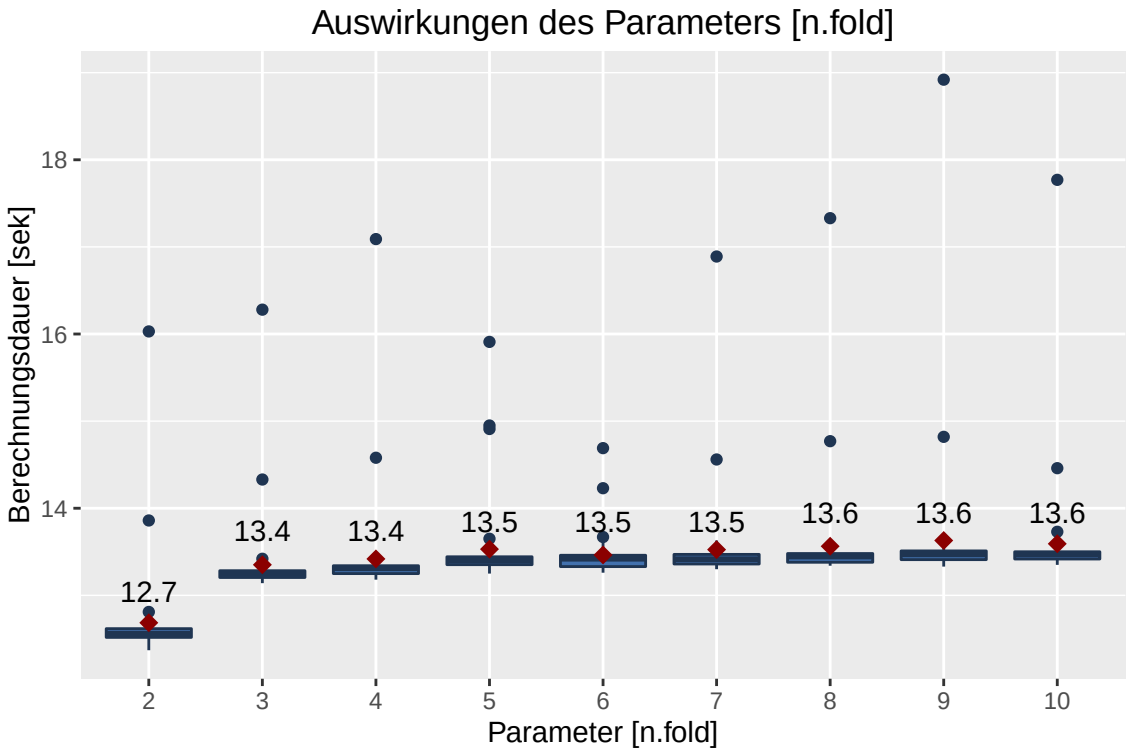

Abbildung 36: Die verschiedenen Werte für den Parameter [`n.fold`] und ihre dazugehörige Berechnungsdauer (Iter.: 1.5). Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

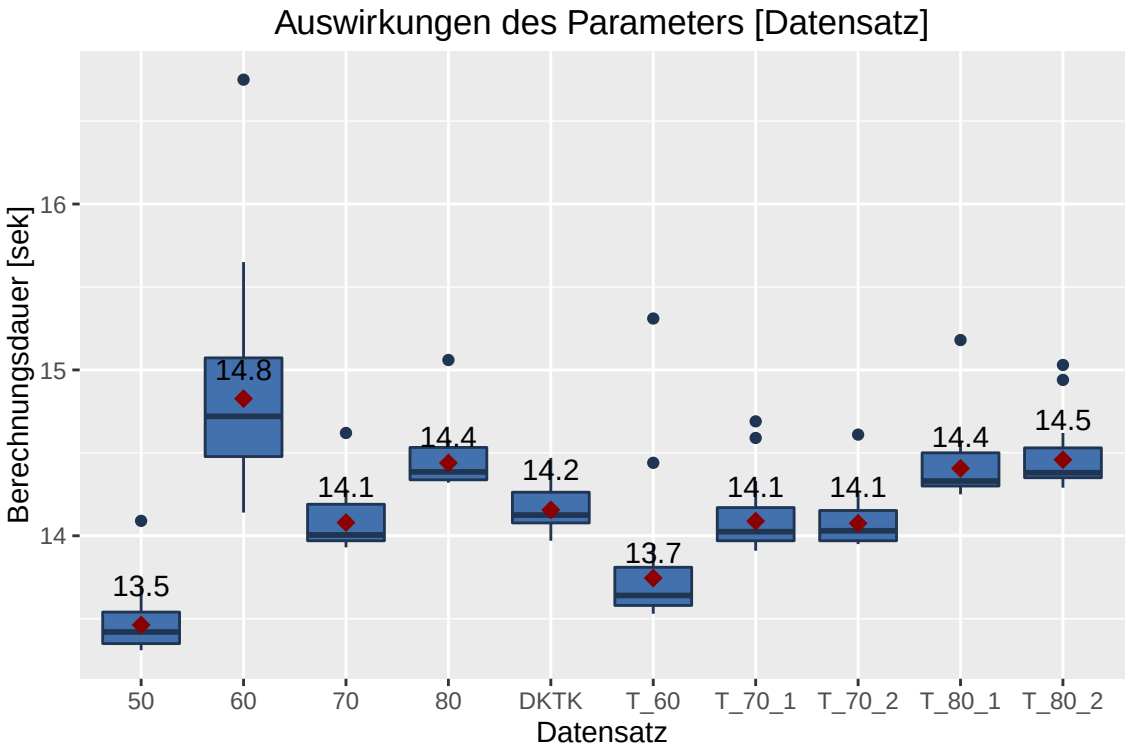

Abbildung 37: Die verschiedenen Trainings-Datensätze und ihre dazugehörige Berechnungsdauer (Iter.: 2.2). Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

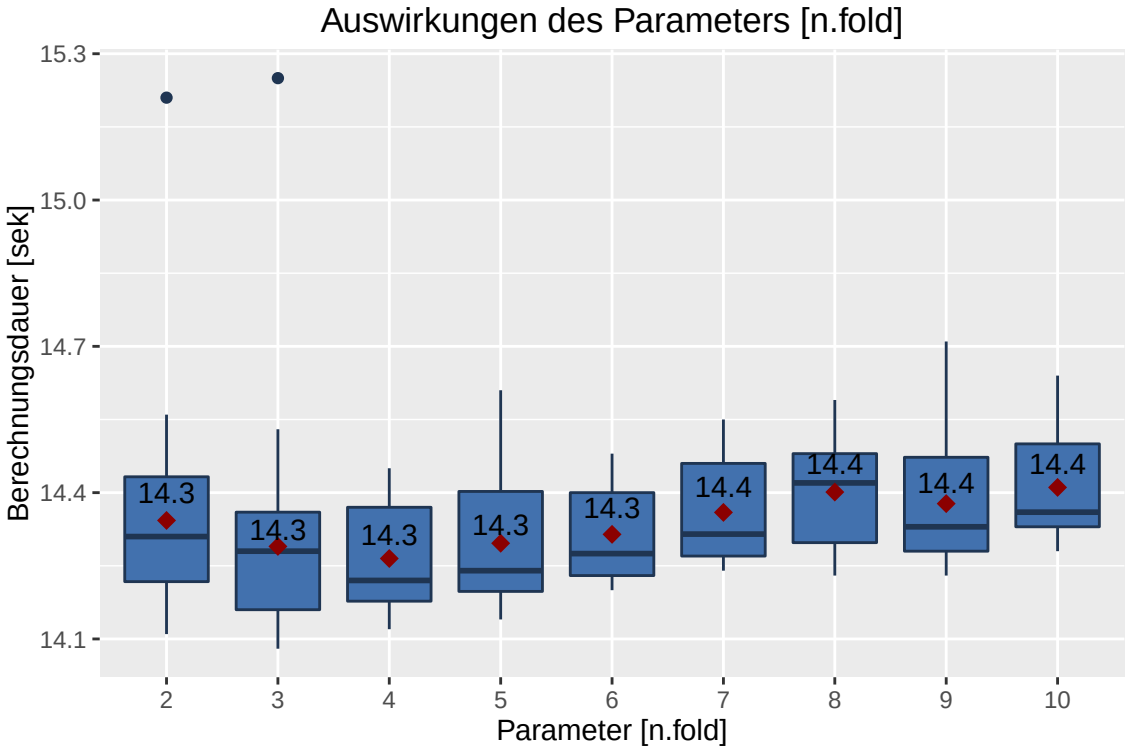

Abbildung 38: Die verschiedenen Werte für den Parameter [`n.fold`] und ihre dazugehörige Berechnungsdauer (Iter.: 2.5). Die roten Punkte sowie die dazugehörigen Zahlen entsprechen den jeweiligen Mittelwerten. Es wurden pro Gruppe 40 Modelle mit unterschiedlichen `seed's` berechnet.

7.2 Digitaler Anhang

Diese CD enthält den digitalen Anhang dieser Bachelorarbeit. Darin befinden sich sowohl alle geschriebenen R-Codes, die verwendeten Abbildungen und die Bachelorarbeit im PDF-Format. Zusätzlich sind weitere Abbildungen vorhanden, die aus Zeitgründen nicht in der Bachelorarbeit verwendet wurden. Mit Hilfe der R-Codes lassen sich zudem weitere Auswertungen und Abbildungen relativ leicht verwirklichen.

Literatur

Borucka, J. (2014). Methods for handling tied events in the cox proportional hazard model. *Studia Oeconomica Posnaniensia*, 2(2):91–106.

Breslow, N. (1974). Covariance analysis of censored survival data. *Biometrics*, pages 89–99.

Calin, G. A. und Croce, C. M. (2006). Microrna signatures in human cancers. *Nature reviews cancer*, 6(11):857.

Cho, H., Yu, A., Kim, S., Kang, J., und Hong, S.-M. (2009). Robust likelihood-based survival modeling with microarray data. *Journal of Statistical Software, Articles*, 29(1):1–16.

Cox, D. R. (1972). Models and life-tables regression. *JR Stat. Soc. Ser. B*, 34:187–220.

Efron, B. (1977). The efficiency of cox's likelihood function for censored data. *Journal of the American statistical Association*, 72(359):557–565.

Engler, D. und Li, Y. (2009). Survival analysis with high-dimensional covariates: an application in microarray studies. *Statistical applications in genetics and molecular biology*, 8(1):1–22.

Fahrmeir, L., Kneib, T., Lang, S., und Marx, B. (2007). *Regression*. Springer.

Gerds, T. A., Kattan, M. W., Schumacher, M., und Yu, C. (2013). Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine*, 32(13):2173–2184.

Gui, J. und Li, H. (2005). Penalized cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. *Bioinformatics*, 21(13):3001–3008.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., und Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546.

Harrell, F. E., Lee, K. L., und Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387.

Kalbfleisch, J. D. und Prentice, R. L. (2002). *The statistical analysis of failure time data*. John Wiley & Sons.

Liu, X. (2012). *Survival analysis: models and applications*. John Wiley & Sons.

Lohaus, F., Linge, A., Tinhofer, I., Budach, V., Gkika, E., Stuschke, M., Balermpas, P., Rödel, C., Avlar, M., Grosu, A.-L., et al. (2014). Hpv16 dna status is a strong prognosticator of loco-regional control after postoperative radiochemotherapy of locally advanced oropharyngeal carcinoma: results from a multicentre explorative study of the german cancer consortium radiation oncology group (dktk-rog). *Radiotherapy and oncology*, 113(3):317–323.

MacFarlane, L.-A. und R Murphy, P. (2010). Microrna: biogenesis, function and role in cancer. *Current genomics*, 11(7):537–561.

Nikulin, M. und Wu, H.-D. I. (2016). *The Cox model and its applications*. Springer.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., et al. (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947.

Shannon, W. D., Watson, M. A., Perry, A., und Rich, K. (2002). Mantel statistics to correlate gene expression levels from microarrays with clinical covariates. *Genetic Epidemiology*, 23(1):87–96.

Witten, I. H., Frank, E., Hall, M. A., und Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wollschläger, D. (2017). *Grundlagen der Datenanalyse mit R: eine anwendungsorientierte Einführung*. Springer-Verlag.

Ziegler, A., Lange, S., und Bender, R. (2004). Überlebenszeitanalyse: Die cox-regression. *DMW-Deutsche Medizinische Wochenschrift*, 129(S 3):T1–T3.

Eigenständigkeitserklärung

Hiermit erkläre ich, Christian Reinhold Bihl, dass ich die vorliegende Bachelorarbeit eigenständig ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort,Datum	Unterschrift