Analyse und Charakterisierung regulatorischer Vorgänge in *Bacillus licheniformis*

Dissertation zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
„Doctor rerum naturalium“
der Georg-August-Universität Göttingen

im Promotionsprogramm Biologie
der Georg-August University School of Science (GAUSS)

vorgelegt von

Sascha Dietrich

aus Hildesheim

Göttingen, 2014

Betreuungsausschuss:

Prof. Dr. Rolf Daniel, Genomische und Angewandte Mikrobiologie, Institut für Mikrobiologie und Genetik

Prof. Dr. Edgar Wingender, Abteilung für Bioinformatik, Universitätsmedizin

Dr. Heiko Liesegang, Genomische und Angewandte Mikrobiologie, Institut für Mikrobiologie und Genetik

Mitglieder der Prüfungskommission:

Referent: Prof. Dr. Rolf Daniel, Genomische und Angewandte Mikriobiologie, Institut für Mikrobiologie und Genetik

Korreferent: Prof. Dr. Edgar Wingender, Abteilung für Bioinformatik, Universitätsmedizin

Weitere Mitglieder der Prüfungskommission:

Prof. Dr. Burkhard Morgenstern, Abteilung für Bioinformatik, Institut für Mikrobiologie und Genetik

Dr. Oliver Valerius, Projektleiter Proteomik, Institut für Mikrobiologie und Genetik

Prof. Dr. Kai Heimel, Abteilung für allgemeine Mikrobiologie, Institut für Mikrobiologie und Genetik

PD Dr. Wilfried Kramer, Abteilung für Molekulare Genetik, Institut für Mikrobiologie und Genetik

Tag der mündlichen Prüfung: 14.01.2015

Inhaltsverzeichnis

Inhalt

Inhaltsverzeichnis

Inhaltsverzeichnis

Abbildungsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

DNA	d*esoxyribonucleic acid*
RNA	r*ibonucleic acid*
mRNA	*messenger* RNA
sRNA	*short* RNA
NGS	*next generation sequencing*
UTR	*untranslated region*
TSS	*transcriptional start site*
QRT-PCR	*quantitative realtime polymerase chain reaction*
BWT	Burrows-Wheeler *transformation*
BLAST	*basic local alignment search tool*
ORF	*open reading frame*
RPKM	*reads per kilobase of transcript per million mapped reads*
FPKM	*fragments per kilobase of transcript per million mapped reads*
NPKM	*nucleotide activites per kilobase of transcript per million mapped reads*
TPM	*transcripts per million mapped reads*
PWM	*positional weight matrix*
HMM	*Hidden-Markov model*
GMM	*general Markov model*
CM	*covariance model*
TPP	*thiaminepyrophosphate*
SAM*	*S-adenosyl-methionine*
SAM*	*Sequence Alignment/Map*
FMN	*riboflavin*
MEME	*Multiple Expectation maximization for Motif Elicitation*
TDS	*transcriptome data set*
COG	*clusters of orthologous groups*
SVG	*scaleable vector graphic*
GFF	*general feature format*
XML	*extensible markup language*

* Bedeutung abhängig vom Kontext

Abkürzungsverzeichnis

JSP	*java server pages*
HTML	*hypertext markup language*
Abb.	Abbildung
z.B.	zum Beispiel
idR.	in der Regel
d.h.	das heißt

1 Einleitung

B. licheniformis DSM13 (Veith *et al.*, 2004)ist ein nicht-pathogener Organismus, der sich durch seine Fähigkeit, große Mengen an Enzymen zu sekretieren, auszeichnet. Dies macht den Organismus zu einer guten Produktionsplatform in der industriellen Produktion von Waschmitteln (Schallmey *et al.*, 2004). Um den Prozess zur Gewinnung von Enzymen zu optimieren, ist ein genaues Wissen über die transkriptionellen Aktivitäten des Organismus sowie der Regulation der zugrundeliegenden Gene notwendig, so dass durch gezielte Manipulation der Gene oder ihrer Regulation der Ertrag bei der Fermentation gesteigert werden kann. Neben der klassischen Form der Regulation auf Proteinebene, wo die Aktivität eines Proteins durch bestimmte Stoffe oder andere Proteine reguliert wird, wurde gerade in den letzten 10 Jahren eine Vielzahl von regulatorischen Effekten beschrieben, die auf Ebene der RNA stattfindet (Mattick, 2004). Somit gewinnt die RNA zunehmend an Bedeutung als Untersuchungsobjekt für das Verständnis der Biologie von Organismen. Die breite Analyse dieser Regulationsebene wurde durch die Entwicklungen der *Second Generation* Sequenzierung ermöglicht, welche ausreichende Sequenzierleistung für diese Aufgabe liefert (van Dijk *et al.*, 2014). Mit dieser hohen Sequenzierleistung geht ein entsprechend hohes Datenaufkommen einher sowie die Notwendigkeit, bioinformatische Methoden einzusetzen und zu evaluieren. Die in dieser Arbeit primär verwendete Methode ist das sogenannte *mapping,* bei dem Sequenzen aus einer RNA-Sequenzierung (RNA-Seq) gegen ein Referenzgenom verglichen werden. Das Ziel hierbei ist es, den Ursprungsort einer RNA-Sequenz (*read*) auf dem Genom zu finden und sie dieser Position zuzuordnen. Durch die Verarbeitung vieler solcher *reads* ist es möglich, die Aktivität von genomischen Bereichen zu analysieren und somit einen Fingerabdruck der vom Organismus verwendeten Gene zu erhalten. Die Verarbeitung solcher Sequenzierdaten und die anschließende Anwendung und Erstellung von Werkzeugen zur Analyse dieser Daten ist die Primäraufgabe der angewandten Bioinformatik.

Die Bioinformatik ist ein interdisziplinäres Feld der Wissenschaft innerhalb derer biologische Fragestellung mit Hilfe von informatischen Methoden bearbeitet werden. Diese Fragestellungen können vielfältig sein und reichen von simplen Verwaltungsaufgaben von Daten bis hin zu komplexen Modellierungen biologischer Funktionen und Strukturen. Die Bioinformatik lässt sich grob in zwei Felder teilen, die algorithmische Bioinformatik, welche sich vor allem mit der Modellierung von biologischen Gesetzmäßigkeiten beschäftigt und die angewandte Bioinformatik, welche diese Algorithmen und Modelle auf biologische Fragestellungen anwendet und evaluiert. Die angewandte Bioinformatik agiert damit an der Schnittstelle zwischen algorithmischer Bioinformatik und klassischer Biologie.

Bioinformatik findet überall dort Anwendung wo biologische Daten aufgrund von Größe oder Komplexität nicht mehr manuell von Menschen verarbeitet werden können. Die Sequenzierung der genomischen Sequenz eines Organismus sowie die Entschlüsselung der Funktionen, welche in diesem Genom kodiert sind, sind klassische Anwendungsgebiete der Bioinformatik. Das klassische Beispiel solcher Funktionen sind die proteinkodierenden Bereiche des Genoms, genannt Gene. Die Gene und ihre Produkte, ob Protein oder funktionale RNA, bestimmen die Fähigkeiten eines Organismus und sind bei der Analyse eines Genoms das Hauptaugenmerk. Für diese Vorhersagen werden bioinformatische Methoden verwendet. Da diese Methoden nicht 100% Genauigkeit erreichen können, ist eine Evaluation durch die Experten des jeweiligen biologischen Gebiets unumgänglich, welche die bioinformatischen Vorhersagen mit biologischem Wissen kombiniert um die Vorhersagen zu bestätigen oder zu verwerfen.

1.1 Motivation

Mit der Entwicklung der *Next Generation Sequencing* Technologien (van Dijk *et al.*, 2014) können alle RNAs einer Kultur in ausreichender Qualität und Menge sequenziert werden. Diese RNA-Seq genannte –omics Technologie ermöglicht eine globale Analyse von Transkriptomen (Narberhaus, 2009). Diese Methode generiert große Mengen an Rohdaten welche hohe Anforderungen an die Werkzeuge zur Auswertung dieser Daten stellen. Momentane bioinformatische Werkzeuge (*tools*) sind in ihrer Verarbeitung von RNA-Seq Experimenten stark durch die Menge der Daten eingeschränkt. Derzeitig verwendete *tools* konzentrieren sich in der Regel auf die im Genom vorhandenen Annotationen um mit Hilfe der RNA-Seq Daten die Aktivität von Genen zu bestimmen. Dies schränkt die Möglichkeiten von Biologen ein, die Daten zu evaluieren und limitiert die Analysen auf die bereits bekannten Annotationen. Solche *tools*, die eine Visualisierung erlauben sind oft Aufgrund ihrer Datenstrukturierung auf wenige Datensätze beschränkt und konzentrieren sich in ihrer Auslegung auf die bereits bekannten Eigenschaften eines Genoms (*features*).

Diese Arbeit ist Teil eines Kooperationsprojekts zwischen der Georg-August Universität Göttingen, und der Henkel KgA, einem Industriepartner. Innerhalb dieses Projekts laufen zwei weitere Doktorarbeiten (A und B), welche mit dieser Doktorarbeit verknüpft sind. Die Doktorarbeit A beschäftigt sich mit der Etablierung einer Fermentation von *Bacillus licheniformis* DSM13 unter den von der Industrie verwendeten Bedingungen. In der Arbeit A wurde außerdem das Protokoll zur Aufreinigung und Sequenzierung der RNA aus der Fermentation erstellt und somit die Datengrundlage für diese Arbeit erzeugt. Das Ziel dieser Arbeit ist die Entwicklung eines *tools* das, neben der Betrachtung der Genaktivitäten, Methoden bereitstellt, um bisher unbekannte *features* des Genoms zu entdecken. Dabei soll das *tool* die Möglichkeit bieten, die Daten visuell zu inspizieren um so Biologen zu

ermöglichen ihr Expertenwissen bei der Suche nach neuen *features* einzubringen. Dies gilt insbesondere für die Doktorarbeit B innerhalb dieses Projekts, welche, auf Basis der Vorhersagen dieser Doktorarbeit, vorhergesagte *features* experimentell untersuchen und verifizieren soll. Diese Aufgabenstellung macht es notwendig, dass das *tool* möglichst viele, im besten Fall alle, Datensätze aus der Doktorarbeit A gleichzeitig bearbeitet um eine große Menge an Bedingungen und Replikaten abdecken zu können. Ein solches *tool* sollte somit den Suchbereich für *features* erweitern und die Möglichkeit eröffnen, die regulatorische Ebene bei der Transkription zu analysieren. Bekannte Klassen solcher *features* sind z.B. untranslatierte Regionen (UTR), Erweiterungen der mRNAs am 5‘ oder 3‘ Ende in denen oft regulatorische *features* enthalten sind, sowie Transkripte ohne proteinkodierende Bereiche, welche funktionale RNAs sein können, die durch Bindung an mRNAs oder Proteine ihre Wirkung entfalten.

1.2 Problemstellung und Lösungsansatz

Die Aufgabe dieser Arbeit lässt sich in zwei Teilbereiche einteilen. Der erste Teilbereich ist die Vorbereitung, Verarbeitung und Speicherung der in 1.1 sowie 2.3.1 beschriebenen RNA-Seq Daten. Der zweite Teilbereich ist die Visualisierung und Analyse der Daten. Für den ersten Teilbereich stehen verschiedene *tools* zur Verfügung, welche Aufgaben wie das *mapping* übernehmen. Diese *mapper* sind sehr effizient und generieren standardisierte Ergebnisformate, welche für die weiteren Schritte verwendet werden können. Für die Speicherung der Daten sowie der *mappings* eignen sich Datenbanken wie z.B. PostgreSQL. Der zweite Teilbereich erfordert die Entwicklung eines eigenen *tools* zur Visualisierung und Analyse, welches die *tools* aus dem ersten Teilbereich verwenden kann. Aufgrund der Größe der betrachteten Daten und da verschiedene Personen mit diesen Daten gleichzeitig arbeiten sollen, soll ein zentralisierter Ansatz für dieses *tool* verfolgt werden. Das Ziel hierbei ist, die Daten auf einem *server* zu belassen und zu verarbeiten und bei den Benutzern nur die Ergebnisse und die Interaktionsoberfläche wiederzugeben, sodass die Menge an zu übertragenden Daten möglichst gering gehalten wird. Auf diese Weise wird auch die Gefahr von Asynchronität zwischen Datensätzen auf seiten der Benutzer verhindert, da alle Daten stets an einer zentralen Stelle verwaltet werden. Dies lässt sich am besten mit einem webbasierten *tool* realisieren, welches auf dem *server* mit den RNA-Seq Daten laufen soll. Eine Herausforderung bei diesem Ansatz ist die Handhabung der RNA-Seq Daten. Die von den *mappern* generierten Ergebnisse sind sehr detailliert und verbrauchen dementsprechend viel Arbeitsspeicher oder Rechenleistung beim Verarbeiten. Um dieses Problem zu lösen müssen die Daten komprimiert und umstrukturiert werden, sodass der Speicherbedarf gering bleibt um so den Einsatz auf einem *server*, der möglichst viele Benutzer bedienen soll, zu ermöglichen.

2 Grundlagen

Die Grundlage der hier aufgeführten bioinformatischen Analysen und die dafür entwickelten *tools* ist die Möglichkeit, das Transkriptom eines Organismus in seiner Gesamtheit zu sequenzieren. Das Transkriptom ist die Gesamtheit aller RNA, die ein Organismus zu einem bestimmten Zeitpunkt gebildet hat. Die Transkriptomsequenzierung übersetzt die RNA zurück in DNA welche dann sequenziert wird. Die Analyse des Transkriptoms erlaubt detaillierte und quantifizierbare Einblicke in die Genexpression eines Organismus mit großer Genauigkeit (Ansong *et al.*, 2013).

Die Nutzung dieser Methode verlangt die Möglichkeit, die erhaltenen Transkriptomsequenzen in ihrem genomischen Kontext zu betrachten und zu evaluieren. Dieser Vorgang wird *mapping* genannt und ist die Grundlage für weiterführende Analysen wie die Expressionsanalyse oder die Suche nach Kandidaten für regulatorische RNAs (Mortazavi *et al.*, 2008; Wang *et al.*, 2009). Neben dem Finden von regulatorischen RNAs erlaubt das RNA-Seq Mapping auch die Identifikation von Startpunkten der Transkription und ermöglicht damit die Suche nach Promotor-Sequenzen die diese Transkriptionsstarts bewirken. Nicolas *et al.* verwenden dieses Vorgehen um in *Bacillus subtilis* Transkriptionsstarts zu bestimmen (Nicolas *et al.*, 2012).

2.1 Transcriptomics

Als Transkriptom bezeichnet man die komplette gebildete RNA eines Organismus zu einem bestimmten Zeitpunkt. Dabei ist zu beachten dass das Transkriptom abhängig ist von den Bedingungen, die im und um den Organismus herrschen, da Organismen ihre Gene in der Regel abhängig von den herrschenden Wachstumsbedingungen transkribieren (Lewin, 2008). Dies wird durch regulatorische Mechanismen bewerkstelligt, die zu verschiedenen Zeitpunkten der Transkription wirken. Solche Regulatoren können Proteine wie σ-Faktoren oder auch RNA Elemente wie *riboswitches* sein. Die RNA-Polymerase, das Enzym welches die DNA abliest und in RNA übersetzt, benötigt zum Ablesen einen sogenannten σ-Faktor, ein Protein welches die Bindung zu spezifischen Mustern, die sogenannten Promotoren in der DNA vermittelt und so die RNA-Polymerase an den Beginn der Transkription navigiert (Sonenshein *et al.*, 2002). Organismen haben mehrere solche σ-Faktoren, welche meist spezifisch für bestimmte Bedingungen oder Funktionen sind, wie z.B. Zellhüll- oder Antibiotikastress (Staroń *et al.*, 2009). Oft sind die σ-Faktoren selber unter der Kontrolle von anderen Regulatoren wodurch äußerst komplizierte Regulationsnetzwerke entstehen.

Neben der Bildung der RNA stellt der Abbau von RNA ein weiteres wichtiges regulatorisches Element dar (Lehnik-Habrink *et al.*, 2012). Die Lebensdauer einer mRNA bestimmt die Menge an Proteinen, die von dieser gebildet werden können. Die Lebensdauer wird

bestimmt durch die Angreifbarkeit der mRNA durch RNAsen. RNAsen sind Enzyme, die RNA zu Mononukleotiden abbauen, wobei es eine ganze Reihe von RNAsen für verschiedene Aufgaben gibt (Deutscher, 1988; Linder *et al.*, 2014). Die Stabilität einer RNA ist daher durch Mechanismen bestimmt, die den Zugang solcher RNAsen kontrollieren, wie z.B. das Vorhandensein von Phosphatgruppen am 5'Ende oder Faltungsstrukturen am 3'Ende der RNA. Ein anderer, suggerierter Faktor ist die Bindung von Ribosomen an mRNA, welche durch ihre Bindung die RNA vor den RNAsen schützen (Belasco and Higgins, 1988). Für die RNA-Seq Methodik bedeutet dies, dass die sequenzierten RNAs stets von verschiedenen Transkripten in unterschiedlichen Degradationsstadien stammen.

Durch die verbesserte Datengrundlage mittels der NGS Methoden wurden bioinformatische Analysemethoden ermöglicht, die quantitative, komparative Ansätze verfolgen und so eine bezüglich der Sensitivität und Spezifität verbesserte Vorhersage von regulatorischen RNAs erlauben (Burge *et al.*, 2013; Eddy and Durbin, 1994). Bei solchen vergleichenden Ansätzen werden manuell kurierte Beispielsequenzen von regulatorischen RNAs gesammelt und miteinander auf ihre konservierten Merkmale verglichen. Aus diesen Vergleichen werden Modelle erstellt, die dann Klassen von regulatorischen RNAs beschreiben und zur Identifikation von neuen Exemplaren dieser Klasse genutzt werden können. Eine andere Strategie ist ein globaler Ansatz, bei dem die transkriptionelle Aktivität eines Organismus mit seinem Genom korreliert wird. Dieser Vorgang wird Transkriptomanalyse genannt. Dabei werden sequenzierte RNA-Fragmente bestimmten *loci* des Genoms zugeordnet, sodass man transkriptionell aktive Bereiche des Genoms von inaktiven Bereichen unterscheiden kann. Diese aktiven *loci* können mit den Annotationen der proteinkodierenden Bereiche des Genoms abgeglichen werden. Über die den Genen zugeordneten RNA-Fragmente können Aktivitätsbestimmungen gemacht werden, was den Vergleich der transkriptionellen Aktivität der Gene ermöglicht (siehe 2.4). In solchen aktiven Bereichen, in denen keine proteinkodierenden Informationen vorliegen, kann man Kandidaten für regulatorische RNAs oder bisher nicht annotierte Gene erwarten. Bei der Suche nach regulatorischen RNAs basierend auf transkriptionell aktiven Regionen ist aber eine genaue Analyse der Indizien für die Anwesenheit einer regulatorischen RNA wichtig, da z.B. Operonstrukturen UTRs enthalten können die keine regulatorischen Elemente enthalten.

2.1.1 Regulatorische RNAs

1990 wurde erstmalig gezeigt, dass RNA-Strukturen in der Lage sind, Metabolite spezifisch zu binden (Ellington, Andrew D; Szostak, 1990). In 2002 wurden dann die ersten Beispiele für molekülbindende mRNAs gefunden, die Aufgrund dieser Bindung die Translation der mRNA beeinflussen (Winkler *et al.*, 2002). Damit war der Beweis erbracht, dass es mRNAs gibt, die einen intrinsischen Regulationsmechanismus für ihre Translationsrate besitzen. Diese Regulationsmechanismen werden genannt und sind eine Subgruppe regulatorischer RNAs (Tucker and Breaker, 2005). Neben den kovalent an mRNA gebundenen regulatorischen RNAs gibt es regulatorische RNAs, die eigenständige RNA-Moleküle sind und in der Regel keinen translatierbaren Bereich besitzen wie z.B. die 6S-RNA (Trotochaud and Wassarman, 2005). Diese regulatorischen RNAs werden als *small* RNAs oder auch *noncoding* RNAs bezeichnet. Obwohl bereits 1967 erstmalig nachgewiesen (Hindley, 1967), blieb ihre Funktion über lange Zeit unbekannt. Der erste Nachweis für einen regulatorischen Effekt durch eine sRNA gelang 1984, in dem ein RNA-Transkript nachgewiesen wurde, dass durch Anlagerung an eine mRNA die Translation blockiert (Mizuno *et al.*, 1984). Ein Nachweis für eine sRNA, die die Translation erst ermöglicht gelang 2010 (Podkaminski and Vogel, 2010). Diese Art von sRNAs werden auch *antisense* RNAs genannt, da ihr Wirkmechanismus auf sequenzkomplementärer Anlagerung an Teile einer Ziel-mRNA basiert (Bouvier *et al.*, 2008). Neben diesen *antisense* sRNAs die mRNAs als Ziel haben, gibt es auch sRNAs die Proteinaktivitäten regulieren, indem sie diese von ihren eigentlichen mRNA-Zielen titrieren (Trotochaud and Wassarman, 2005). Ein Beispiel für eine solche sRNA ist die 6S RNA, welche einem offenen SigA-Promotor ähnelt und daher RNA-Polymerasen mit SigA σ-Faktoren bindet um so die Expression von SigA Genen zu regulieren (Steuten *et al.*, 2013).

2.1.2 σ-Faktoren

σ-Faktoren sind DNA-Bindeproteine, welche temporärer Bestandteil des RNA-Polymerase Holoenzymkomplexes sind. Sie bewirken eine Bindung dieses Komplexes an spezifische Erkennungsmuster (Promotoren) in der DNA womit sie den Startpunkt der Transkription kontrollieren. Die σ-Faktoren verfügen über zwei Bindedomänen, σ2 und σ4 welche die Interaktionen mit den -10(-12) bzw. -35(-24) *patterns* durchführen (Paget and Helmann, 2003) (siehe 2.1.3). Einige σ-Faktoren verfügen zusätzlich über eine σ3 Domäne, welche für die Interaktionen mit erweiterten -10 *patterns* notwendig ist (Campbell *et al.*, 2002).

Es sind mehrere σ-Faktoren in *Bacillus* beschrieben(Sonenshein *et al.*, 2002; MacLellan *et al.*, 2008; Staroń *et al.*, 2009). Der SigA σ-Faktor wird auch der *housekeeping σ-factor* genannt, da er der σ-Faktor für die allgemein benutzten Gene ist. Neben diesem *housekeeping σ-factor* gibt es verschiedene σ-Faktoren, welche in speziellen Situationen und

Bedingungen benutzt werden. Dies sind z.B. SigB für die allgemeine Stressantwort, SigD für die Transkription von Flagellengenen, SigH für die Sporulation und Ausbildung der Kompetenz, SigE und SigF für die Genregulation in der Mutterzelle während der Sporulation, SigK und SigG für die Genregulation der Vorspore während der Sporulation. Der SigI σ-Faktor ist für dieKontrolle der Zellantwort auf Hitzeschock zuständig (Zuber *et al.*, 2001) während SigL Gene für Reaktion auf Kälteschock kontrolliert (Merrick, 1993; Wiegeshoff *et al.*, 2006). SigL ist der einzige bekannte σ-Faktor der σ^{54} Familie in *Bacillus* während die restlichen σ-Faktor zur σ^{70} Familie gehören.

Zusätzlich gibt es eine spezielle Gruppe von σ-Faktoren, genannt *extracytoplasmatic function σ-factors* (ECF-σ-Faktoren). Die ECF Gruppe unterscheidet sich von anderen σ-Faktoren durch spezifische Eigenschaften. Strukturell besitzen sie keine σ3-Domäne und werden durch Sensorproteine, die Anti-σ Faktoren, kontrolliert, welche die ECF-σ-Faktoren binden. Diese anti-σ-Faktoren werden oft als Operon zusammen mit dem entsprechenden ECF-σ-Faktor transkribiert, wobei nicht alle ECFs einen anti-σ-Faktor aufweisen. Viele dieser Anti-σ-Faktoren sind membranständige Proteine, die auf spezifische, extrazelluläre Signale reagieren und bei Aktivierung den gebundenen ECF-σ-Faktor freigeben wodurch dieser an RNA-Polymerasen binden kann um so die Gene seines spezifischen Regulons zu transkribieren (Staroń *et al.*, 2009).

Eine schematische Darstellung der relevanten Domänen von σ– sowie ECF-σ-Faktoren findet sich bei Staroń *et al.* (Abb. 1).

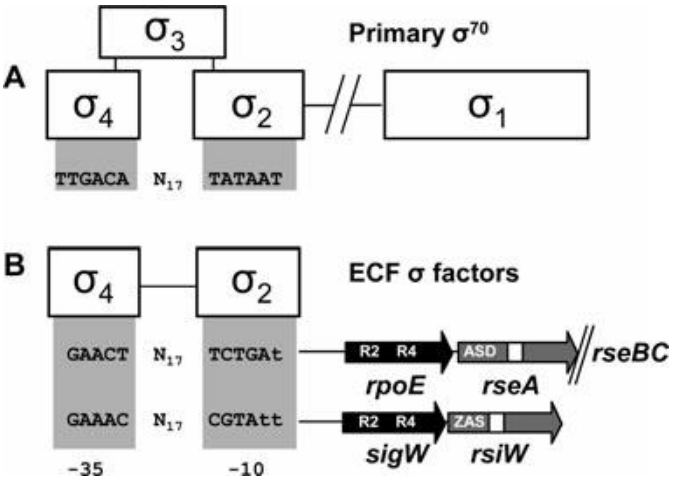

Abb. 1 Schematischer Aufbau von σ-Faktoren und ECF-σ-Faktoren nach Staroń *et al.*

2.1.3 Promotoren

Promotoren sind spezifische Sequenzmuster (genannt *patterns*) im Genom, die RNA-Polymerasen als Signal für die Initiierung der Transkription dienen. Promotoren werden von den spezifischen Untereinheiten der RNA-Polymerase, den sogenannten σ-Faktoren (siehe 2.1.1), erkannt worauf an einer spezifischen Stelle *downstream* vom Promotor die Transkription eingeleitet wird (Sonenshein *et al.*, 2002). Damit sind Promotoren ein entscheidendes Element in der Erkennung von funktionellen Bereichen des Genoms. Ihre eindeutige Identifizierung ist aber eine große Herausforderung in der Bioinformatik (Pedersen *et al.*, 1999; Das and Dai, 2007). Ein Promotor besteht in der Regel aus zwei kurzen *patterns*, die von entsprechenden σ-Faktoren erkannt werden (Lewin, 2008). Diese Bindung dirigiert den RNA-Polymerase Holoenzymkomplex an die Transkriptionstartstelle (TSS) worauf dann die Transkription eingeleitet werden kann. Die bioinformatische Herausforderung bei der Vorhersage von Promotoren basiert darauf, dass Bindestellen auf Sequenzebene nur gering konserviert sind. Untersuchungen an der σ2-Bindestelle des SigA σ-Faktors haben gezeigt, dass nur drei von sechs Basen aus der Bindestelle tatsächlich konserviert sind (Feklistov and Darst, 2011). Daher werden bei Promotoren Konsenssequenzen definiert, die die Schnittmenge aller bisher bekannten Promotorsequenzen eines Typs darstellen. Reelle Promotorsequenzen weichen in der Regel von diesem Konsensus ab. Diese Abweichungen scheinen eine wichtige Eigenschaft für die Regulation der Promotoren zu sein, da eine zu hohe Ähnlichkeit oder Abweichung zum Konsens die Transkriptionsrate reduziert (Ellinger *et al.*, 1994). Variation ist demnach ein Merkmal von Promotoren. Abb. 2 zeigt ein LOGO (Crooks *et al.*, 2004) erstellt von Jarmer *et al*. (Jarmer *et al.*, 2001) basierend auf bioinformatischen Vorhersagen von SigA Promotorsequenzen in *Bacillus subtilis* 168.

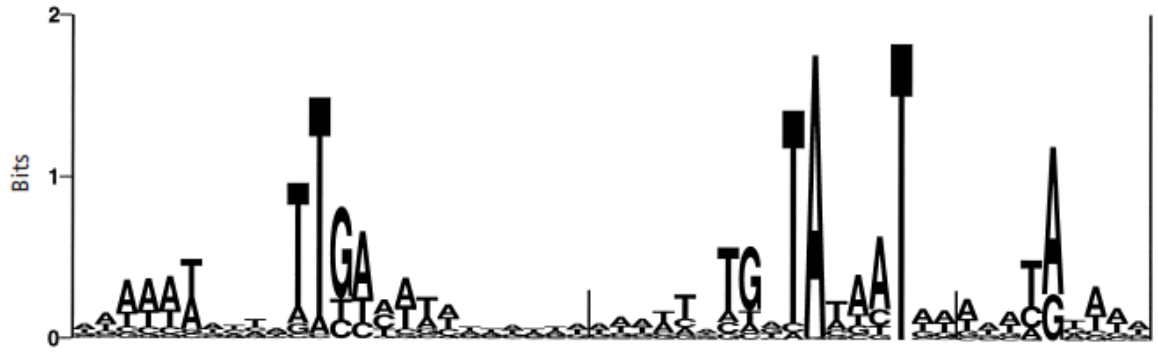

Abb. 2: LOGO des SigA Konsensus nach Jarmer *et al*.
Die Höhe der Buchstaben repräsentiert den Grad der Konserviertheit im pattern.

Staroń *et al*. haben WebLogos für verschiedene ECF-σ-Faktorbindestellen erstellt (Abb. 3).

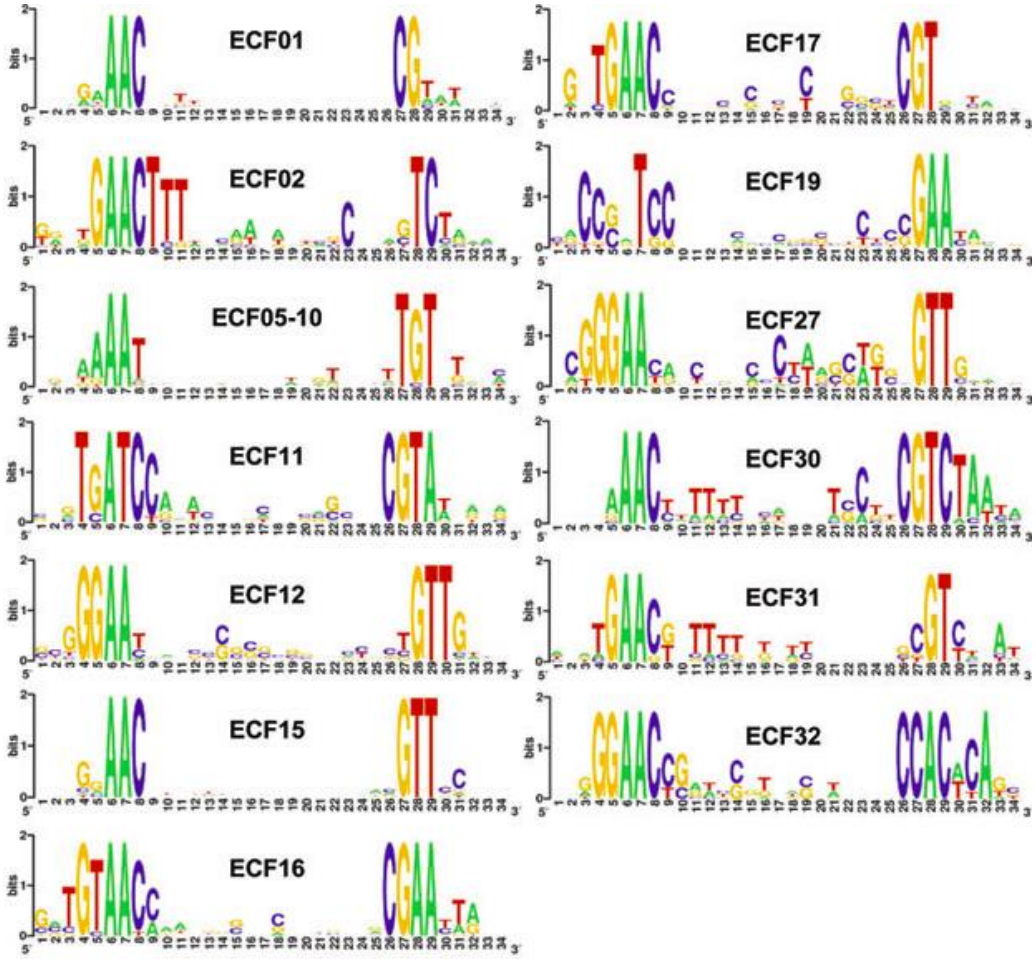

Abb. 3: WebLogo Promotor *patterns* verschiedener ECF *σ*-Faktoren nach Staroń *et al*.
Die Höhe der Buchstaben repräsentiert den Grad der Konserviertheit im pattern.

Anhand von Abb. 2 und Abb. 3 ist die Variabilität der Promotorsequenzen gut erkennbar. Auch lässt sich in Abb. 3 gut erkennen, dass manche der ECFs sehr ähnliche *patterns* verwenden was die Zuordnung von spezifischen Bindestellen erschwert. Aufgrund dieser Ähnlichkeiten kann es zu Kreuzaktivitäten zwischen verschiedenen ECFs kommen (Mascher *et al.*, 2007).

Der SigA sowie die ECF-σ-Faktoren gehören zur Familie der σ^{70} Sigma-Faktoren (Sonenshein *et al.*, 2002). Diese Sigma-Faktoren erkennen Promotorsequenzen an den -10 und -35 Positionen relativ zum Startpunkt der Transkription. Eine weitere Familie von Sigma-Faktoren, σ^{54} (σ^{N}), beschrieben in *Salmonella enterica* serovar Typhimurium, erkennen *patterns* an den Positionen -12 und -24 (Sonenshein *et al.*, 2002). Somit ist die Distanz der konservierten *patterns* in Relation zum Transkriptionsstartpunkt ebenfalls ein wichtiges Merkmal für Promotoren.

Weitere Promotor *patterns* sind in der Prodoric Datenbank (Munch, 2003) zu finden, welche *positional weight matrices* (siehe 2.5), für viele dieser *patterns* bereitstellt. Allgemein wird angenommen, dass *B. licheniformis* DSM13, aufgrund seiner phylogenetischen Nähe zu *B. subtilis*, ähnliche oder gleiche Promotorsequenzen für seine σ-Faktoren verwendet und die σ-Faktoren auch die gleichen Funktionen haben. Vergleichende Suchen zwischen *E. coli* und verwandten Enterobakterien zeigen dass anhand von Sequenzvergleichen konservierte Promotor *patterns* identifiziert werden können (Mccue *et al.*, 2001).

2.2 Phylogenie und Biologie von regulatorischen RNAs

Während viele regulatorische RNAs phylogenetisch im vergleichbarem Maß wie Gene konserviert sind (Zhang *et al.*, 2004), hat sich gezeigt, dass diese Konservierung primär die Faltungsstrukturen (hierbei ist intermolekulare Doppelstrangbildung eingeschlossen) betrifft. Die Basensequenz selber, die zwar der Faltungsstruktur zugrunde liegt, ist aber nur in begrenztem Maße konserviert. Allerdings ist die Mutationsrate von Basen in Doppelstrangbereichen mit der Mutationsrate der komplementären Base verknüpft, ein Effekt der als Kovarianz bezeichnet wird (Eddy and Durbin, 1994). Allgemein lässt sich sagen, dass die Faltungsstruktur, da sie das funktionsgebende Element der regulatorischen RNAs sind, das primäre Erkennungsmerkmal von regulatorischen RNAs ist (Lindgreen *et al.*, 2006). Ein zweites, wichtiges Merkmal regulatorischer RNAs ist ihre spezifische Stabilität. Damit regulatorische RNAs ihren Effekt ausüben können, müssen sie in ausreichender Menge in der Zelle vorliegen und ausreichend lange existieren, sodass sie mit ihren Zielen interagieren können. Bei *riboswitches* sind diese Faktoren in der Regel durch ihre zugehörige mRNA bestimmt, wobei es aber Ausnahmen geben kann, wie z.B. selbstschneidende *riboswitches* (Tucker and Breaker, 2005). Die tatsächliche Lebenszeit einer regulatorischen RNA kann stark variieren, bedingt durch ihren kontrollierten Abbau durch eine Vielzahl verschiedener Ribonukleasen (Göpel *et al.*, 2013; Viegas *et al.*, 2007; Davis and Waldor, 2007). Studien zur Sequenzkonserviertheit sind derzeit noch selten. Im Modellorganismus *Escherichia coli* wurde gezeigt, dass die bisher bekannten regulatorischen RNAs in etwa die gleiche Konserviertheit wie Gene aufweisen (Zhang *et al.*, 2004). Untersuchungen an eukaryotischen regulatorischen RNAs haben gezeigt, dass der Konserviertheitsgrad stark von der Länge der regulatorischen RNA und ihrer Lokalisierung innerhalb einer größeren mRNA abhängt. Es zeigte sich, dass längere regulatorische RNAs sich in eher variable und eher konservierte Bereiche aufteilen (Pang *et al.*, 2006).

2.3 Mapping

Die Zuordnung von *reads* aus einem RNA-Seq Experiment erfolgt durch das sogenannte *mapping*. Dafür gängige Algorithmen sind in Ruffalo *et al.* beschrieben (Ruffalo *et al.*, 2011). Beim *mapping* wird eine Suchsequenzen mit einer größeren Referenzsequenz verglichen mit dem Ziel, eine Position in der Referenzsequenz zu finden, die eine möglichst hohe Ähnlichkeit zwischen Such- und Referenzsequenz aufweist. Im Fall des RNA-Seq *mappings* bedeutet dies, die beste Übereinstimmung zwischen sequenzierter RNA, den sogenannten *reads* und einer Genomsequenz zu finden. Bei ausreichender Sequenzähnlichkeit zwischen *read* und Genomsequenz geht man davon aus, dass die sequenzierte RNA von dieser Position im Genom transkribiert wurde. Wird eine Stelle im Genom gefunden, an die der *read* mit ausreichender Ähnlichkeit passt, wird er dieser Position zugeordnet und gilt damit als *mapped.* Das Ergebnis dieses *mappings* ist eine Verteilung der *reads* im Genom, welche die transkriptionelle Aktivität des Organismus widerspiegelt. Diese spezifischen Verteilungen sind das primäre Ergebnis des *mappings* und die Grundlage aller analytischen Methoden die bei der Auswertung der Transkriptomsequenzierung angewandt werden.

Die Grundannahme dass im Falle einer ausreichenden Ähnlichkeit die RNA Sequenz von einer Stelle im Genom transkribiert wurde wird durch repetitive Regionen, sogenannte *repeats* im Genom in Frage gestellt. *Repeats* stellen beim *mapping* ein großes Problem dar, da *reads* die ausschließlich in *repeat*-Bereichen liegen, nicht eindeutig einem Bereich zugeordnet werden können. Beim RNA-Seq *mapping* ist daher eine Sequenziermethode mit möglichst geringer Fehlerrate und möglichst großer *read* Länge wichtig, da für ein eindeutiges *mapping* unter Umständen bereits einzelne Basen wichtig sein können. *Reads* die mehreren *genom loci* mit gleicher Qualität zugeordnet werden können, werden hier als *multimaps* bezeichnet. Derzeit gibt es drei Strategien mit solchen Bereichen umzugehen. (i) *Multimapped reads* können z.B. allen Positionen an denen sie passen zugeordnet werden. Dies führt zu einer künstlichen Verdopplung dieser *reads.* Die Folge ist eine Erhöhung der Abdeckung durch Aufaddierung der *reads* wodurch *repeats* aktiver erscheinen als sie tatsächlich sind. (ii) Die *reads* können gleichmäßig oder nach komplizierteren Schlüsseln auf die Repeatbereiche zu verteilt werden. (iii) *Multimaps* werden bei der Betrachtung von Aktivitäten nicht beachtet. Allen drei Strategien ist gemein, dass die Ansätze zu methodenspezifischen Fehlern in der Beschreibung der transkriptionellen Aktivitäten führen. Im Allgemeinen ordnen *mapper multimaps* stets mehreren Bereichen zu und überlassen die Handhabung dieser Bereiche den Wissenschaftlern und *tools*, die das *mapping*-Ergebnis analysieren.

Allen Mappern gemein ist, dass sie die von ihnen berechneten *mappings* in verschiedenen Formaten speichern können. Das gängigste dieser Formate ist das Sequence

Alignment/Map (SAM, siehe 2.3.3) Format. Über diese Austauschformate wird es möglich, die Mappinginformationen in analytischen Programmen zu verwenden.

2.3.1 RNA-Seq Daten

Mit der Entwicklung der sogenannten *second generation* Sequenziertechnologien (Niedringhaus *et al.*, 2011) (hier definiert als Methoden die auf klonfreier PCR- Amplifikation von einzelsträngigen DNA Fragmenten basieren wie bei der 454 Pyrosequenzierung und der Illumina/Solexa Technologie.) wurde es möglich, strangspezifische Transkriptom-sequenzierungen in ausreichender Menge und Qualität durchzuführen. Diese Technologien eröffnen die Möglichkeit, die transkriptionelle Leistung eines Organismus in seiner Gesamtheit zu beschreiben und damit Einblicke in die Aktivität und Regulation von allen aktiven Genen eines Organismus unter ausgewählten Wachstumsbedingungen zu erhalten. Wo ehemals nur gezielte Einzelexperimente zu Genen mittels QRT-PCR und 5'RACE möglich waren, sind jetzt Ansätze möglich die für alle Gene eines Organismus Daten in vergleichbarer Qualität liefern. Die Datenmengen in modernen Sequenzierexperimenten stellen informatische Auswertungsmethoden vor neue, teils extreme Problemstellungen in Bezug auf den Bedarf an Speicher und Rechenleistung und erfordern damit neue Strategien der Verarbeitung.

RNA-Seq Daten können je nach verwendeter Methode unterschiedlich aussehen. In den meisten Sequenziertechnologien werden die ursprünglichen Transkripte fragmentiert auf eine bestimmte Leselänge, welche je nach Technologie unterschiedlich lang ist. Die Sequenziertechnologien liefern zudem unterschiedliche Qualitäten. Die eingangs erwähnte 454 Pyrosequenzierung generiert z.B. Probleme bei *homopolymer-stretches*, Bereichen in denen die gleiche Base mehrfach hintereinander auftaucht. Die Illumina Technologie hat eine erhöhte Fehlerrate in GC-reichen Bereichen (Aird *et al.*, 2011). Bei beiden Technologien steigt die Fehlerrate mit zunehmender Länge der *reads.* Diese Fehlerquellen erschweren das *mapping* der *reads*.

2.3.2 Bowtie2 und BWA

Bowtie2 (Langmead and Salzberg, 2012) und BWA (Li and Durbin, 2009) sind *mapper,* die die Burrows-Wheeler Transformation (BWT) implementieren um *reads* auf schnelle und speichereffiziente Weise auf eine Referenz zu *mappen*. Die Burrows-Wheeler Transformation ist ein Kompressionsverfahren, das wiederkehrende Zeichen in einer Zeichenkette indexbasiert ordnet. Bioinformatisch können diese *indices* genutzt werden um Subsequenzen in einer größeren Sequenz zu suchen. Die Burrows-Wheeler Transformation (BWT) ist eine *ungapped*, also lückenlose Suchmethode und ist damit erst einmal nicht für

das *mapping* von *reads*, die ja mit Sequenzunterschieden behaftet sein können, geeignet. Bowtie2 und BWA lösen dieses Problem, indem sie nur Teile der *reads* mit der BWT *mapped* und diese Teile dann als *seeds* verwenden, um das Alignment danach mittels dynamischer Programmierung zu vervollständigen.

Bowtie2 kann beim *mapping* lokale wie auch globale Alignments produzieren und eignet sich daher auch für Fälle, in denen Teile von *reads* erwartungsgemäß nicht *mappen* sollten. Bowtie2 generiert als Ergebnis eine SAM formatierte Datei. BWA bietet ähnliche Funktionen wie auch Bowtie2 hat aber zusätzlich die Möglichkeit, längere *reads* wie z.B. aus der 454 Sequenzierung zu bearbeiten. In dieser Arbeit wurde für die *mappings* und die Vergleiche die Version 2.0.0-beta5 von Bowtie2 und die Version 0.6.1-r104 von BWA verwendet.

2.3.3 SAM Dateiformat

SAM steht für Sequence Alignment/Map und ist ein Dateiformat, in dem *mapping* Informationen von *reads* gespeichert werden (Li *et al.*, 2009). Tabelle 1 listet diese Felder mitsamt einer kurzen Beschreibung auf.

Tabelle 1: Felder des SAM Formats

Feld	Name	Beschreibung
1	QNAME	*Query Name*. Name des *reads*
2	FLAG	Bitwise Flag. *Bit-flag* mit Informationen über den Status des *reads* im *mapping.*
3	RNAME	*Reference Sequence*. Name der Referenzsequenz, auf die der *read* gemappt wurde. Bei nicht gemappten *reads* ist dieser Wert ein *
4	POS	1-*Based leftmost Position.* Die linke Anfangsposition des *mappings* im Bezug zur Referenz.
5	MAPQ	*Mapping Quality*. Phred skalierte *mapping quality* Information für gesammten *read.*
6	CIGAR	*CIGAR String*.Beschreibung des Alignments pro Base des *reads.*
7	MRNM	*Mate Reference Name*. Name des *mate reads* bei *paired-end reads*
8	MPOS	1-*Based leftmost Position of Mate.* Die linke Anfangsposition des *mappings* des *mate reads* im Bezug zur Referenz.
9	ISIZE	*Inferred Insert Size.* Größe des Inserts bei *paired-end reads.*
10	SEQ	*Query Sequence*. Sequenz des *reads* in Ausrichtung zur Referenz
11	QUAL	*Query Quality.* Phred basierte Einzelbasenqualitäten des *reads*
12	MISC	*Miscellanious.* Zusätzliche Felder mit Programmspezifischen Informationen

Für die Beurteilung eines *mappings* im Falle von RNA-Seq Experimenten werden die Position, die b*it-flag* sowie der CIGAR String verwendet. Die *bit-flag* enthält Informationen darüber, ob der *read* überhaupt gemappt wurde, ob er in Plus- oder Minus-Richtung in Relation zur Referenz liegt und ob es sich um ein *multimapping* handelt oder nicht. Anhand der *bit-flag* lässt sich also feststellen, ob ein *read* überhaupt genauer betrachtet werden muss. Liegt ein erfolgreich gemappter *read* vor, wird der CIGAR String betrachtet, der Informationen darüber enthält, welche Basen des *reads* zur Referenz gemappt wurden und wo der *mapper* eventuell Insertionen und Deletionen gesetzt hat. Dies geschieht über eine Kodierung aus Buchstaben und Zahlen. Beschrieben wird das *mapping* von links nach rechts wobei immer Bereiche beschrieben werden in Form eines Buchstaben für den Zustand gefolgt von einer Zahl für die Anzahl an Basen. Als mögliche Zustände besitzt der CIGAR String *match/mismatch* dargestellt durch ein `M`, `I` für *Insert,* `D` für *Deletion,* `N` für ausgelassene Basen auf der Referenz, `S` für *soft clipping*, `H` für *hard clipping* und `P` für *padding.* Da im CIGAR String nicht zwischen *match* und *mismatch* unterschieden wird, benötigt man zur Feststellung der Anzahl der *mismatches* einen zusätzlichen Wert. Dieser sollte vom *mapper* nach dem Standard in den zusätzlichen Parametern einfügt werden. Dieser Wert, `NM:i:`, gibt dann die Anzahl an Unterschieden zwischen *read* und Referenzsequenz an.

2.3.4 Mapping mit BLAST

BLAST (Altschul SF, Gish W, Miller W, Myers EW, 1990), basic local alignment search tool, ist ein gängiges *tool* zum Suchen von Sequenzen in Datenbanken. Bei der BLAST Suche wird eine Suchsequenz in *seeds* aufgeteilt, die dann in der Datenbank gesucht werden. Sobald ein *seed* gefunden wurde, wird ausgehend von diesem *seed* ein Needleman-Wunsch Alignment durchgeführt um ein lokales Alignment zu erhalten. Obwohl es eigentlich ein Suchprogramm ist, kann BLAST auch für ein *mapping* verwendet werden. Dazu muss man die erhaltenen BLAST Treffer, die ja lokale Alignments sind, in ihrer Länge in Bezug zur Gesamtlänge des *reads* setzen. So kann man BLAST Treffer verwerfen die nicht eine ausreichende Länge des *reads* betreffen. Zusätzlich muss man Grenzwerte für *mismatches* definieren um eine ausreichende Qualität des *mappings* sicherzustellen. Beide Informationen liefert BLAST in seinen Ergebnissen und kann somit auch zum *mappen* von RNA-Seq Experimenten benutzt werden. Wurtzel *et al.* haben RNA-Seq Experimente mittels BLAST *mappings* erfolgreich durchgeführt (Wurtzel *et al.*, 2010). Die in dieser Arbeit verwendete Version des BLASTs ist Version 2.2.18.

2.4 Vergleich von Expressionsstärke

Bei der Transkriptomsequenzierung wird mittels NGS Technologien die gesamte RNA eines Organismus sequenziert. Das Ergebnis sind, nach *mapping*, die *mapped reads* welche dann in weiteren Analysen verwendet werden. Die Menge an *reads* ist jedoch abhängig vom Sequenzieransatz. Daher kann die Menge an sequenzierten *reads* für ein bestimmtes *feature* nicht als Aktivitätswert benutzt werden. Stattdessen wird die Menge an *reads* für ein *feature* ins Verhältnis zur Gesamtmenge an *mapped reads* gesetzt. Dadurch wird der Einfluss des Sequenzieransatzes normalisiert und die erhaltenen Werte können als Wert für die Expressionsstärke, d.h. als Maß für die Aktivität eines *features* unter bestimmten Bedingungen, verwendet und verglichen werden. Ein solcher Vergleich von Expressionsstärken wird differentielle Expressionsanalyse genannt, wo die normalisierte Expressionsstärke eines *features* unter verschiedenen Bedingungen miteinandern verglichen wird. Die Anzahl der *mapped reads* unterliegt zwei Arten von Bias. Technischer Bias entsteht bei unterschiedlichen Mengen an sequenzierter RNA bei verschiedenen Experimenten. Dieser Bias wird durch die oben beschriebene Normalisierungsmethodik entfernt. Methodischer Bias ist stets spezifisch für die verwendete Sequenziertechnologie (siehe auch 2.3.1). Im Falle der für die Daten dieser Arbeit verwendeten Illumina Sequenziertechnologie ist dies ein Längen-Bias, der lange gegenüber kurzen Genen bevorzugt. Dieser Bias entsteht durch die Aufbereitung der *library*, wo nur Fragmente einer bestimmten Länge für die Sequenzierung akzeptiert werden. Längere Gene haben beim *shearing* statistisch gesehen eine höhere Wahrscheinlichkeit, ein akzeptables Fragment zu generieren als kurze (Li *et al.*, 2010).

Eine Methode zur Berechnung von normalisierten Expressionsstärken sind RPKM-Werte (r*eads per kilobase of transcript per million mapped reads*) (Mortazavi *et al.*, 2008). Die RPKMs beziehen sich immer auf einen Bereich des Genoms und dienen dem Vergleich von Expressionsstärken ohne aufwendige statistische Methoden verwenden zu müssen. RPKMs normalisieren die Expressionswerte gegen die Gesamtanzahl an *mapped reads* im Experiment und stellen damit die Vergleichbarkeit zwischen verschiedenen Sequenzierexperimenten her. Sie sind aber z.B. durch den von Li *et al.* beschriebenen methodischen Bias begrenzt.

Neben den RPKM gibt es FPKMs (*Fragments per kilobase of transcript per million mapped fragments*) (Trapnell *et al.*, 2010). Sie sind analog zu RPKMs, abstrahieren aber von einzelnen *reads* und zählen cDNA Fragmente. Diese FPKMs werden z.B. in *paired-end* Sequenzierungen verwendet, wo zwei *reads* zu einem Fragment vereint werden. Für FPKMs gelten die gleichen Limitierungen wie RPKMs.

Um den methodischen Bias von RPKMs zu umgehen, wurden TPMs (*Transcripts per million mapped reads*) definitiert (Wagner *et al.*, 2012). Wagner et *al.* beschreiben eine Methode die Aktivitätswerte berechnet, die nicht vom Bias durch die Leselänge der Sequenziertechnologie betroffen sind. Dabei wird nicht direkt gegen die Anzahl der *mapped reads* wie bei den RPKMs normalisiert, sondern gegen die Anzahl an Transkripten. Die Anzahl an Transkripten pro Gen wird aus den *mapped reads* berechnet und dann gegen die Größe des jeweiligen Gens normalisiert, sodass der Bias der Sequenziertechnologie gegen kürzere Gene aufgehoben wird. Dieser Ansatz ist damit abhängig von der Qualität der Annotation.

Genauere Methoden der Expressionsanalyse wie baySeq (Hardcastle and Kelly, 2010) and DESeq (Anders and Huber, 2010) verwenden statistische Modelle und erreichen damit eine höhere Verlässlichkeit in ihren Aussagen, haben jedoch den Nachteil des höhere Rechenaufwands, der mit solchen Analysen einhergeht.

Allen Methoden der Normalisierung ist gemein, dass für eine aussagekräftige Expressionsanalyse die Abdeckung ausreichend hoch sein muss, da sonst unterschiedliche Expressionsstärken nicht mehr oder nur unzureichend aufgelöst werden können (Liu *et al.*, 2013).

2.5 Pattern finding

Unter *pattern finding* versteht man die Suche nach Sequenzmustern in biologischen Sequenzen (DNA oder Aminosäuresequenzen). Die Komplexität kann hier je nach verwendeter Methode stark variieren. Die einfachste Variante ist eine Suche mittels Sequenzvergleich wie z.B. BLAST, welche einen direkten Vergleich zwischen Such- und Referenzsequenz macht. Daher erlaubt die BLAST-Suche nur ein begrenztes Maß an Variation und ist somit ungeeignet, stark variierende Muster zu suchen. Um Muster mit Variationen zu beschreiben, werden sogenannte *positional weight matrices,* kurz PWMs (Levitsky *et al.*, 2007; Li *et al.*, 2007), benutzt. PWMs werden durch ausgesuchte Beispielsequenzen gebildet, wobei an jeder Position des PWMs die Häufigkeit einer jeden Base festgehalten wird. Anhand dieser Häufigkeiten kann dann eine Suchsequenz gegen die PWM verglichen werden und ein *score* berechnet werden, der wiedergibt, wie gut die Suchsequenz zur PWM passt. PWMs können keine Insertionen und Deletionen beschreiben. Außerdem ist jede Position in der PWM unabhängig wodurch Abhängigkeiten in der Folge der Positionen nicht beschrieben werden können. PWMs werden vor allem in der Suche nach Promotorbindestellen verwendet. Hidden-Markov-Modelle, kurz HMMs (Eddy, 1996), definieren Abfolgen beliebiger Elemente durch die Betrachtung der Häufigkeit des Auftretens eines Elements in Abhängigkeit von den vorangegangenen Elementen. Sowohl Nukleotid- als auch Proteinsequenzen stellen solche Abfolgen aus Elementen dar und lassen sich so

beschreiben. Betrachtet man die Gesamtheit der möglichen Ketten die HMMs beschreiben, bilden diese einen Baum. Jeder Knoten dieses Baumes beinhaltet für diese Position spezifische Wahrscheinlichkeiten für das Auftreten eines Ereignisses. Beim Beispiel von Nukleotid Sequenzen wären das die Wahrscheinlichkeiten für das Auftreten von einzelnen Basen. Zusätzlich zu diesen Informationen können HMMs auch Insertionen und Deletionen sowie Wiederholungen von Mustern unterschiedlicher Länge beschreiben. Um ein HMM zu erstellen, benötigt man Beispielsequenzen, die miteinander *aligned* werden müssen. Aus diesem *multiple alignment* wird dann das HMM generiert. HMMs eignen sich zur Suche von Proteinsequenzen. Sie sind aber nicht in der Lage, Abhängigkeiten der einzelnen Positionen über mehrere Positionen hinweg abzubilden. Solche Abhängigkeiten werden aber für die Beschreibung von Faltungsstrukturen von RNAs benötigt. Um diese Faltungsstrukturen zu beschreiben, werden Kovarianzmodelle (CMs, siehe 2.5.1) verwendet. Tabelle 2 gibt einen Überblick über die verschiedenen Methoden und deren Anwendungsbereiche.

Tabelle 2: Übersicht über die verschiedenen *pattern finding* Methoden und ihrer Anwendungsgebiete

Programm/Modell	**Anwendungsgebiet**
BLAST	Direktvergleich von zwei Sequenzen
Positional Weight Matrices	Suche nach Sequenzmustern definierter Länge mit Variationen. Beispiel Promotoren- und Bindestellenmotive.
Hidden-Markov-Modelle	Suche nach Sequenzmustern mit variabler Länge und Variationen. Beispiel Proteindomänen.
Kovarianzmodelle	Suche nach Sequenzmustern variabler Länge die abhängige Variationen (Kovarianzen) enthalten. Beispiel gefaltete RNAstrukturen.

2.5.1 Kovarianzmodelle

Kovarianzmodelle sind eine Variante von HMMs (Eddy and Durbin, 1994). Sie beschreiben sowohl die primäre Basensequenz einer sRNA wie auch die Paarung einzelner Basen mit anderen Basen innerhalb der Sequenz wodurch die Faltungsstruktur representiert wird. Die Modelle beschreiben dabei einen Baum, bei dem jeder Knoten ein mögliches Ereignis darstellt. Diese Ereignisse können die Baseninteraktion der entsprechenden Basen in der RNA sein. Interaktionen können hier Paarungen mit anderen Basen oder *singlet* Basen sein, die nicht gepaart vorliegen. Weitere Ereignisse sind Insertionen, Deletionen und Bifurkationen, die Basenunabhängig sind. Jedes dieser möglichen Ereignisse hat eine Wahrscheinlichkeit, mit der es in der jeweiligen Folge auftreten kann. Diese Wahrscheinlichkeiten werden anhand von multiplen a*lignments* von bekannten Sequenzen,

den sogenannten *seed* Sequenzen, der gleichen Art von regulatorischer RNA errechnet in dem in diesen multiplen *alignments* mittels dynamischer Programmierung die maximale Anzahl an Paarungsevents der Basen zwischen allen Spalten des *alignments* bestimmt wird. Da für die Bildung einer Struktur in der Regel nicht die Identität einer Base in der Struktur wichtig ist, sondern die Paarbildung mit ihrem Partner in der Struktur, kann es sein dass im Fall einer Mutation nicht die mutierte Base revertiert, sondern der Bindungspartner in der Struktur eine komplementäre Mutation vollzieht. Dieser Effekt koppelt die Wahrscheinlichkeit des Auftretens einer Base an die Wahrscheinlich des Auftretens der korrespondierenden Base und wird als Kovarianz bezeichnet. Durch die Erfassung dieser Kovarianzen lässt sich ein Modell erstellen, das die Paarungen der einzelnen Basenpositionen und damit die Faltungsstruktur der RNA beschreibt.

Programme wie Infernal (siehe 2.5.2) benutzen diese Modelle um damit RNA-Sequenzen zu identifizieren, die sich dem Modell entsprechend falten können. Bei dieser Suche wird ein *score* für die Suchsequenz errechnet (Nawrocki *et al.*, 2009). Will man diesen evaluieren, vergleicht man ihn mit Kovarianzmodell Grenzwerten, auch *cutoff* Werte genannt. Jedes Modell besitzt spezifische *cutoff* Werte, die bei der Erstellung des Modells ermittelt werden. Der erste solche *cutoff* Wert ist der *trusted cutoff*. Dieser Wert ist der geringste *score*, den eine *seed* Sequenz, aus der das Modell erstellt wurde, gegen sein eigenes Modell erreicht. Annahme ist hierbei, dass alle Sequenzen, deren *score* oberhalb dieser Grenze liegt, zum Modell gehören. Damit sind sie ein Feature, das von diesem Modell beschrieben wird. Hat man mehrere Features, die phylogenetisch nahe verwandt sind und damit eine Gruppe bilden, kann man den *gathering cutoff* bestimmen. Der *gathering cutoff* ist der geringste *score*, den eine Sequenz der nahe Verwandten gegen das Modell erreicht. Die dritte Art von *cutoff* ist der *noise cutoff,* der ein Maß dafür darstellt, wie spezifisch ein Modell ist. Um den *noise cutoff* zu bestimmen, werden zufällig generierte Sequenzen gegen das Kovarianzmodell verglichen und der höchste *score* wird als *noise cutoff* definiert. Der *noise cutoff* dient der Abschätzung, wie sehr ein Modell auf Rauschen (im Sinne von zufälligen Sequenzen ohne biologische Relevanz) reagiert. Der *noise cutoff* sollte bei Modellen möglichst niedrig sein.

2.5.2 Rfam

Rfam ist eine 2003 veröffentlichte Datenbank für Kovarianzmodelle von regulatorischen RNAs. Die Datenbank bietet Zugang mittels eines WWW-Interface auf seine Datenbestände. Die regulatorischen RNAs werden in Familien geordnet, zu denen ein Kovarianzmodell gepflegt wird. Für alle Kovarianzmodelle sind die ursprünglichen *seed* sequenzen verfügbar sowie die ermittelten *cutoff* Werte. Rfam bietet neben den Kovarianzmodellen auch Referenzen zu den einzelnen Familien sowie eine integrierte Suchmethode für Sequenzen.

Diese Suchmethode ist aber limitiert in der Anzahl der Suchsequenzen. Für die Suche mit vielen Sequenzen bietet Rfam die Kovarianzmodelle zum Herunterladen an sodass auf lokalen Servern diese aufwendigen Suchen durchgeführt werden können. Zur Verwendung der Rfam Modelle wird die Programmsammlung Infernal benötigt (Nawrocki *et al.*, 2009). Die Programme der Infernalsuite dienen dem Erstellen von und der Suche mit Kovarianzmodellen. Die in dieser Arbeit verwendete Version der Rfam Datenbank ist die Version 11.0, die verwendete Infernal Version ist 1.0.2.

2.5.3 MEME

MEME steht für *Multiple Expectation maximization for Motif Elicitation*. Der MEME Algorithmus sucht in einer Gruppe von biologischen Sequenzen nach gemeinsamen Motiven innerhalb dieser Sequenzen. Dabei wird nach sogenannten durchgängigen (contiguous) Motiven gesucht, das heißt die Motive dürfen Punktmutationen aber keine Insertionen oder Deletionen beinhalten. Der MEME Algorithmus benutzt dabei einen modifizierten EM (Expectation maximization) Algorithmus, um aus Teilabschnitten der Eingangssequenzen möglichst optimal konservierte Motive zu finden (Bailey, 1995), wobei die Länge der Motive vorgegeben sein muss. Die Teilsequenzen werden dann miteinander verglichen und *weight matrices* für die einzelnen Basen errechnet. Basierend auf diesen *weight matrices* wird eine Kombination an Teilsequenzen gesucht, die über ein möglichst hohes Gewicht der einzelnen Basen in den jeweiligen Positionen verfügt. Das Ergebnis des EM Algorithmus ist eine Position pro Sequenz, ab der mit der höchsten Wahrscheinlichkeit ein Motif der gesuchten Länge liegt, das mit allen anderen Sequenzen geteilt wird. Der EM Algorithmus selber setzt voraus, dass jede Sequenz ein passendes Motif beinhaltet. Dies ist aber bei biologischen Sequenzen, insbesondere bei einer Sammlung von Promotorsequenzen verschiedener σ-Faktoren, nicht zwangsweise gegeben. Der MEME Algorithmus umgeht diese Limitation indem er heuristische Merkmale bei der Bewertung der *weight matrices* benutzt um Sequenzen, die nicht über ein passendes Motif verfügen, aus den Eingabesequenzen für den EM Teil zu filtern. Desweiteren ist MEME in der Lage, systematisch verschiedene Startpunkte von Motiven in einer Sequenz zu überprüfen um so das mehrmalige Vorkommen eines Motifs zu erkennen. Das Ausschließen von Sequenzen für die Eingabe, bei zu geringer Wahrscheinlichkeit des Motifs, erlaubt es MEME Rauschen durch unpassende Sequenzen zu verringern. Die in dieser Arbeit verwendete MEME Version ist Version 4.9.0.

3 Komparative Identifikation von regulatorischen RNAs

Die rein bioinformatische Identifikation von regulatorischen RNAs ist eine große Herausforderung (Backofen and Hess, 2010). Wie in 2.2 erwähnt, stellt die hohe Veränderbarkeit der zugrundeliegenden RNA-Sequenz, unter Beibehaltung der funktionellen Struktur, bioinformatische Ansätze vor große Probleme und schränkt reine Basenvergleiche basierend auf Homologie als Suchmethode stark ein (Eddy *et al.*, 1994). Verschiedene Ansätze existieren, die andere Merkmale der regulatorischen RNAs als Suchkriterium verwenden. Eine Übersicht über die Grundlegenden Ansätze haben Backofen und Hess zusammengestellt, für Details siehe (Backofen and Hess, 2010). Kontextspezifische Suchen konzentrieren sich auf den genomischen Kontext, in dem eine regulatorische RNA vorkommen sollte. *riboswitches* z.B. sollten stets in der genomischen Nachbarschaft bestimmter Gene auftreten. Damit lässt sich der Suchbereich für bestimmte regulatorische RNAs eingrenzen. Ein weiteres Merkmal ist die stabile Faltungsstruktur von regulatorischen RNAs. Programme wie RNAz (Washietl *et al.*, 2005) können benutzt werden um das Potential zur Bildung stabiler Sekundärstrukturen von RNAs zu bestimmen. Auch wenn regulatorische RNAs in ihrer Sequenz stark variieren können, sind ihnen durch die Notwendigkeit der Konservierung ihrer Funktion Grenzen in ihrer Veränderbarkeit gesetzt (Eddy and Durbin, 1994). Dies ermöglicht es, regulatorische RNAs durch komparative Vergleiche zwischen sehr nahe verwandten Organismen zu finden. All diesen Überlegungen ist gemein, dass sie für sich genommen nur mäßige Erfolge erzielen. Durch eine Kombination dieser Methoden ist es jedoch möglich, durch Abgleich ihrer Ergebnisse die Genauigkeit bei der Suche nach regulatorischen RNAs zu steigern (Tjaden, 2008).

Im Rahmen meiner Diplomarbeit wurde eine Methode in Zusammenarbeit mit Christian Opitz und Isabelle Heinemeyer entwickelt, die regulatorische RNAs über die Konserviertheit intergenischer Regionen in nahe verwandten Organismen identifiziert. Die Methode, genannt sRNAfinder (namensgleich zum sRNAfinder von Brian Tjaden (Tjaden, 2008)), reduziert Genome auf ihre intergenischen Bereiche und aligniert diese miteinander. In solchen intergenischen Regionen, die mindestens zwischen drei Organismen aligniert werden konnten, wird dann eine RNAz (Gruber *et al.*, 2010) Analyse durchgeführt, um potentielle stabile RNA-Strukturen zu identifizieren. Diese Suche nach stabilen RNA-Strukturen dient als Vorfilter für den weitaus zeitaufwendigeren Schritt der Suche mit Kovarianzmodellen. Annahme hierbei ist, dass regulatorische RNAs auch stabile RNA-Strukturen enthalten und diese somit als Filterkriterium verwendet werden können. Wurden konservierte RNA-Strukturen gefunden, werden diese Bereiche einer Analyse mittels Infernal unter Verwendung der Rfam-Modelle unterzogen um mögliche regulatorische RNAs zu identifizieren.

Die sRNAfinder Methode von Brian Tjaden verfolgt einen ähnlichen Ansatz, fügt aber zusätzliche Merkmale wie Promotoren und Terminatoren hinzu und definiert *general Markov models* (GMMs) welche den generellen Aufbau eines Operons beschreiben. Über diese GMMs werden dann die Bereiche im Genom identifiziert, die in den zeitaufwendigen Analyseschritten auf regulatorische RNAs überprüft werden.

In meiner Diplomarbeit wurde die sRNAfinder Methode erfolgreich auf *Bacillus licheniformis* DSM13 angewandt. Insgesamt konnten 47 verschiedene strukturelle RNAs vorhergesagt werden worin essentielle sRNAs wie z.B. die 6S-RNA, tmRNA und die RNAseP enthalten waren. Neben diesen essentiellen sRNAs wurde auch eine Vielzahl von *riboswitches* identifiziert. Diese Vergleiche waren erfolgreich, da gut annotierte Genome von nahe verwandten Organismen verfügbar waren. Nach Abgleich dieser 47 Vorhersagen mit den TraV Kandidatenlisten (siehe 6) konnten für 43 von den Vorhersagen Kandidaten in TraV gefunden werden, welche diese in den *loci* einschließen.

In Kooperation mit Beatrix Suess wurde die sRNAfinder Methode auf *Streptomyces coelicolor* A3 angewandt (Vockenhuber *et al.*, 2011). In dieser Arbeit wurden intergenische sRNAs mittels *Deep-Sequencing* und Northern-Blots nachgewiesen und bestätigt. Die Vorhersagen des sRNAfinders wurden mit diesen experimentell bestätigten Vorhersagen verglichen. Durch diesen Vergleich zeigte sich, dass Vorhersagemethoden basierend auf phylogenetischer Konserviertheit stark abhängig von den verfügbaren Vergleichsorganismen sind. Beim Vergleich der RNAz Vorhersagen konnten 208 von 1252 sRNAs identifiziert werden. Nur 31 dieser Treffer konnten mit einem Rfam Modell als bereits bekannt bestätigt werden wobei die Zuordnung bei den meisten Treffern fragwürdig ist, da *microRNA* Modelle diese Treffer lieferten (siehe HitsInf.xls und HitsRNAz.xls auf der Daten-CD unter den sRNAfinder Vorhersagen). Dieser Versuch zeigt die Limitation der phylogenetischen Suche nach regulatorischen RNAs, da nahe Verwandtschaft der Vergleichsorganismen sowie eine hohe Qualität der Vergleichsgenome Voraussetzung für erfolgreiche Suchen sind. Des Weiteren ist die Methode der Identifizierung durch Rfam abhängig von der Qualität der Kovarianzmodelle. Organismengruppen, für die nur wenige oder keine Beispielsequenzen für regulatorische RNAs vorhanden sind, sind bei der Suche mit Rfam benachteiligt. Tabelle 3 gibt einen Überblick über einige Modelle, die Treffer in *B. licheniformis* DSM13 produziert haben. Die Modelle für TPP- und SAM-*riboswitches* haben dabei verlässliche Treffer oberhalb des *trusted cutoff* generiert. Sie stellen Beispiele für Modelle mit hoher Qualität dar. Die Modelle für *bsrG* und das ROSE Element verfügen nur über eine geringe Anzahl an Beispielsequenzen. Beide Modelle generieren Treffer in *Bacillus licheniformis* DSM13 welche aber unterhalb des *trusted cutoff* liegen. Im Falle des *bsrG* kann man annehmen, dass eine

solche regulatorische RNA in *B. licheniformis* DSM13 vorkommen kann, da es sich um eine *Bacillus* spezifische regulatorische RNA handelt. Das ROSE Element ist ein thermosensitiver *riboswitch* der ausschließlich in Alphaproteobakterien beschrieben ist (Chowdhury *et al.*, 2003). Daher ist es unwahrscheinlich, dass in *B. licheniformis* DSM13 ein ROSE Element existiert. Beide Modelle zeigen dass bei geringer Datengrundlage die Genauigkeit der Kovarianzmodelle stark abnimmt.

Tabelle 3: Beispiele für Rfam Modelle und ihrer Abundanz im Genus *Bacillus*

Rfam Modell	**Anzahl Beispielsequenzen im Genus *Bacillus***	**Anzahl an Beispielsequenzen**	**Seed Größe für das CV-Modell**
TPP *riboswitch* (RF00059)	420	11197	115
SAM *riboswitch* (RF00162)	841	4757	433
bsrG (RF01412)	42	172	6
ROSE (RF00435)	0	111	13

Dieser Vergleich der sRNAfinder Methode mit den Ergebnissen einer auf experimentellen Daten basierenden sRNA Suche belegt, dass *Deep-Sequencing* Ansätze die weitaus sensiblere Suchmethode darstellen.

4 Mapping von RNA-Seq Daten auf *Bacillus licheniformis* DSM13

Als Datengrundlage dieser Arbeit dienen die gemeinsam mit Sandra Wiegand publizierten (Wiegand *et al.*, 2013) experimentellen Ergebnisse. Diese Arbeit hatte die Erstellung von RNA-Seq Datensätzen in einer industriellen Fermentation von *Bacillus* licheniformis DSM13 zum Ziel. Es wurden fünf verschiedene Proben im Verlauf der Fermentation genommen mit jeweils drei Replikaten. Zusätzlich zu diesen 15 Datensätzen wurden von den fünf Probezeitpunkten fünf Proben einer 5‘ phosphatabhängigen Exonuklease (TEX) Behandlung unterzogen. Diese Behandlung dient der Anreicherung von RNA-Molekülen an denen die Transkription initiiert wurde(Vockenhuber *et al.*, 2011). Innerhalb der Fermentation wurde die sporulationsdefiziente Mutante MW3 (ΔspoIVA) (Waschkau *et al.*, 2008) des *B. licheniformis* Typstamm DSM13 verwendet. Dies ist notwendig, da in Fermentern keine Organismen benutzt werden dürfen die keimfähige Sporen bilden können. Alle weitergehenden Untersuchungen wurden auf Sequenzdaten und Annotationen des DSM13 durchgeführt. Fermentationsstamm spezifische Deletionen stellen für die folgenden Analysen kein Problem dar, da die Stämme ansonsten isogenisch sind. Diese Deletionsbereiche wurden als interner Qualitätsstandard verwendet. Abb. 4 zeigt die Deletion im *hsdR* Gen in MW3. Das *mapping* der MW3 RNA-Seq Daten auf DSM13 führt zu fehlender Abdeckung der in MW3 deletierten Bereiche im DSM13 *mapping*. Die Grenzen der Transkripte in Abb. 4 passen genau zur von Waschkau *et al.* beschriebenen Deletion.

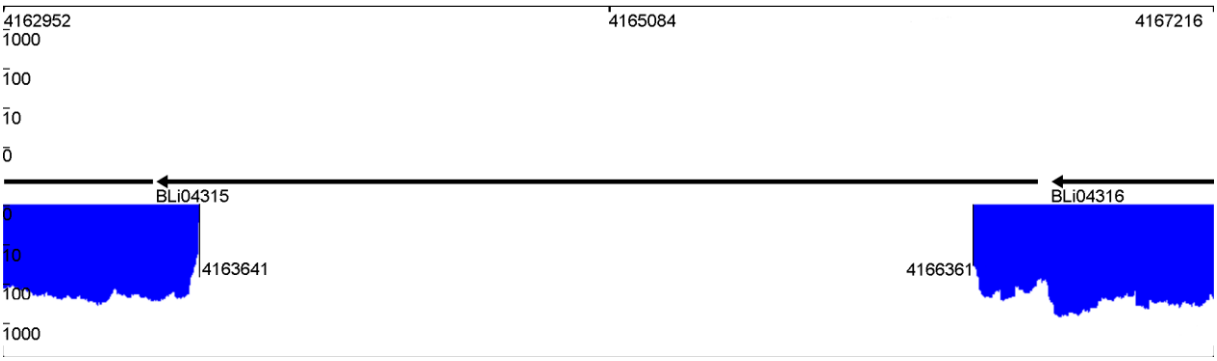

Abb. 4: Deletierter Bereich im Gen *hsdR* (BLi04315)
Das *mapping* der MW3 RNA-Seq Daten auf DSM13 zeigt den in MW3 deletierten Bereich durch fehlende Abdeckung auf. Der fehlende Bereich umfasst 2720 Basen was der Deletion wie von Waschkau *et a*l. beschrieben entspricht

Das erste Ziel der Arbeit war es, eine verlässliche Mappingstrategie für die erhobenen Daten zu finden. Zu diesem Zweck wurde ein *tool*, genannt RNAseqMapper, entwickelt, welches BLAST für das *mapping* verwendet (siehe 2.3.4). Damit ein *read mapped* wird, darf er maximal einen *mismatch* aufweisen und muss über die volle Länge des *reads* erfolgreich *mapped* sein. Bei der im Experiment verwendeten Readlänge von 50 Basen entspricht das minimal einer 98% Ähnlichkeit zwischen *read* und Genomsequenz. Zusätzlich zu dieser Ähnlichkeit muss der *read* ein eindeutiges *mapping* aufweisen, d.h. es darf nur eine Stelle im

Genom geben an der der *read* am besten passt. Diese Parameter entsprechen einer sehr konservativen Vorgehensweise. Um die Verlässlichkeit dieser Methode zu überprüfen wurden zusätzlich zwei weitere *mapper,* Bowtie2 und BWA angewandt. Für beide *mapper* wurden die Standardparameter der jeweiligen Programme verwendet mit Ausnahme der Anzahl an *mismatches* innerhalb ihrer *seeds*. Da für den RNAseqMapper ein *mismatch* erlaubt wurde, dürfen Bowtie2 und BWA im *seed* jeweils auch einen *mismatch* aufweisen.

Für Bowtie2 und BWA wurde das *tool* SAMtoTDS entwickelt, das es erlaubt die gleichen Qualitätskriterien wie beim RNAseqMapper anzuwenden und die Mappingergebnisse für die späteren Analysen in das TDS Format umwandelt. Dieses *tool* nutzt die in 2.3.3 beschriebenen Informationen des SAM Dateiformates um die einzelnen *read mappings* zu evaluieren. Zu diesem Zweck werden die *bitflag*, *CIGAR string* und *mismatch* Informationen verwendet. Die *bitflag* wird ausgewertet um zu bestimmen, ob der *read* überhaupt als *mapped* betrachtet wurde. Von solchen *reads*, die vom *mapper* als *mapped* betrachtet wurden, wird dann der *CIGAR string* ausgewertet. Hierbei wird überprüft, ob Bereiche des *reads* geschnitten (*clipped*) wurden. Diese geschnittenen Positionen werden standartmäßig als *mismatches* betrachtet, können aber mittels Laufzeitparametern auch entfernt werden so dass für einen verkürzten *read* ein *mapping* entsteht. Zu diesen *mismatches* durch das *clipping* werden dann die *mismatches* aus dem eigentlichen *alignment* hinzugezählt und die prozentuale Ähnlichkeit zwischen Gesamtlänge des *reads* und Referenz berechnet und gegen den bei Laufzeit angegebenen *cutoff* (in dieser Arbeit 98%) verglichen. Ist die prozentuale Ähnlichkeit größer oder gleich dem *cutoff* wird der *read* als *mapped* akzeptiert und gespeichert. Kommt im Verlauf der Auswertung der gleiche *read* nochmals vor, wird diese Auswertung für die zweite Position wiederholt und dann mit den Werten des ersten *mappings* verglichen. Bei gleicher Qualität des *mappings* wird der *read* als *multimapped* markiert, ansonsten wird das *mapping* mit der besseren Qualität für den *read* behalten und das schlechtere *mapping* verworfen. Ein *read,* der *multimapped* ist, kann nur durch ein besseres *mapping* wieder zu einem eindeutigen *mapping* werden. Nachdem alle *reads* des *mappings* prozessiert wurden, werden für jede Position im Genom die Anzahl an *mapped reads* auf dem Plus- und Minus-Strang gezählt und im TDS Format gespeichert. Die einzelnen *read mapping* Informationen werden danach verworfen.

Zur Veranschaulichung werden die Anzahl der *mapped*, *unmapped* und *multimapped reads* zwischen den verschiedenen *mappern* in Abb. 5 grafisch dargestellt. In Abb. 6 wird zudem die Laufzeit der *mapper* dargestellt.

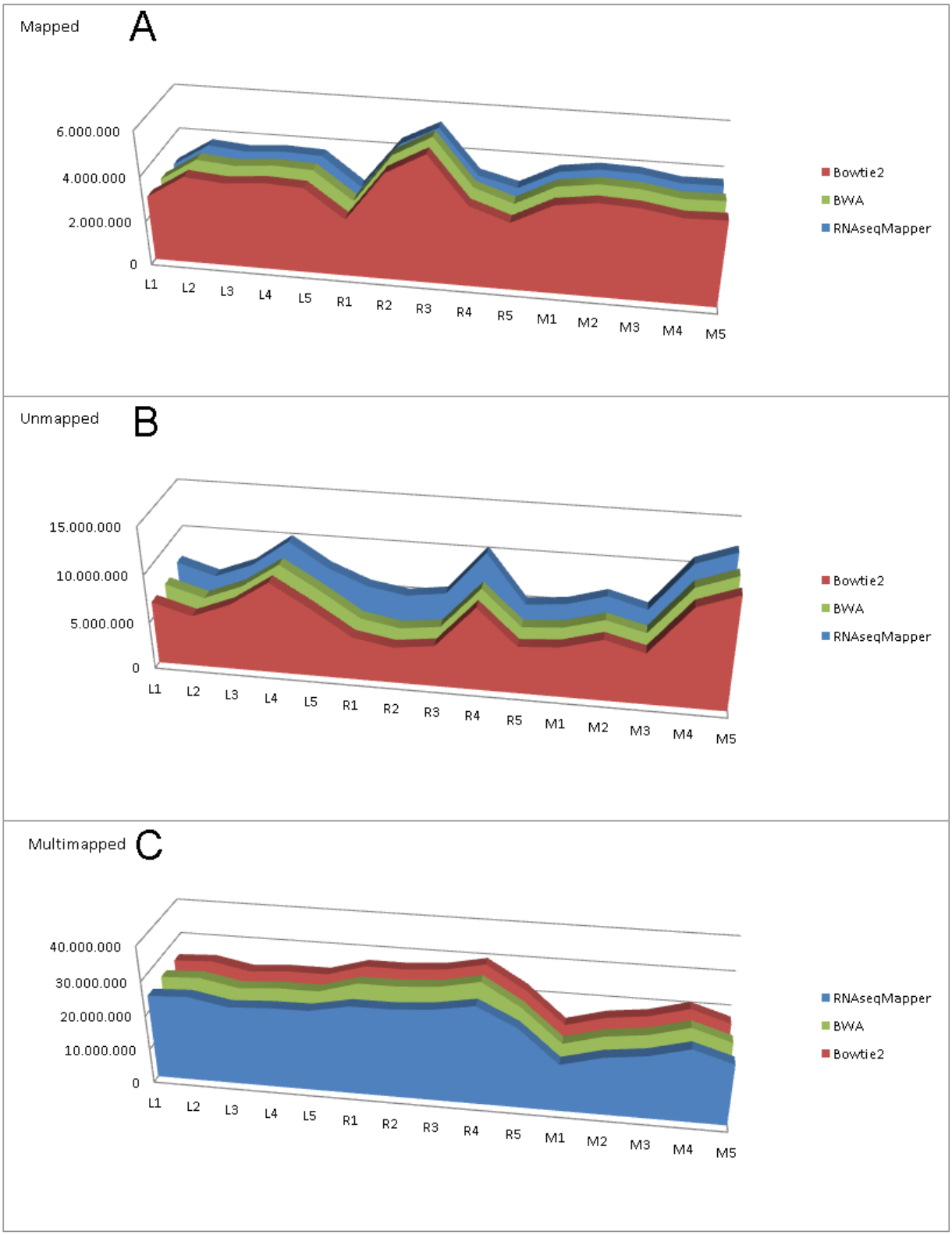

Abb. 5: Ergebnisse des *mappings* der RNA-Seq Daten auf *B .licheniformis* DSM13
Die X-Achse zeigt die Namen der Datensätze, die Y-Achse die Anzahl an *reads.* Grafik A zeigt den Graphen für die Anzahl der *mapped reads*, Grafik B den Graphen für die Anzahl der *unmapped reads*, Grafik C zeigt den Graphen für die Anzahl der *multimapped reads*

Anhand der Abb. 5(A) lässt sich zeigen dass die Qualität des *mappings* in Bezug auf die *mapped reads* nahezu gleich ist. Unterschiede zwischen BWA und Bowtie2 bewegen sich in

der Größenordnung von ca. 1000 *reads*. Der Unterschied zwischen den BWT basierten *mappern* und dem RNAseqMapper liegt in der Größenordnung von ca. 150,000 *reads*, was etwa einer Menge von ~5% der jeweiligen Anzahl an *mapped reads* entspricht. RNAseqMapper und die BWT basierten mapper zeigen leichte Unterschiede in der Menge der *unmapped* und *multimapped reads*, siehe Abb. 5(B) und Abb. 5(C) sowie Tabelle 4. Die drei verglichenen *mapper* unterscheiden sich in der Anzahl ihrer gemappten *reads* im Bereich von unter einem Prozent. Größere Unterschiede, im Bereich von ~1 bis 5% zeigen sich nur in der Anzahl an *multimapped reads* und *unmapped reads.* Da diese *reads* für die spätere Auswertung nicht verwendet wurden, kann man sagen dass die drei *mapper* in der Qualität ihrer Ergebnisse gleichwertig sind.

Tabelle 4: Übersicht der prozentualen Verteilung der *reads* zwischen den drei Mappern

	mapped reads	***multimapped reads***	***unmapped reads***
RNAseqMapper	6,9% - 15,9%	51,8% - 74,9%	15% - 36,5%
Bowtie2	7,5% - 16,6%	52,9% - 78,8%	10,6% - 34,9%
BWA	7,5% - 16,6%	52,9% - 78,9%	10,6% - 34,9%

Um zu überprüfen, ob die Verteilung der *reads* vergleichbar ist, wurde die prozentuale Abdeckung der Gene durch die *mapped reads* verglichen (siehe Liste der Prozentunterschiede auf der Daten-CD im Verzeichnis MappingComparison). Da das *mapping* in repetitiven Bereichen wie in 2.3 erwähnt oft fehlerbehaftet ist, wurden Gene, die über solche Bereiche verfügen, von der Analyse ausgeschlossen. Ein Gen wurde als unterschiedlich abgedeckt betrachtet wenn die prozentuale Abdeckung zwischen mindestens zwei *mappern* größer als fünf Prozent ist. Im Falle unserer Datensätze bewegt sich die Anzahl an unterschiedlich abgedeckten Genen im Bereich von 1,3 bis 5,6 Prozent aller Gene in *B. licheniformis* DSM13. Damit ist gezeigt, dass die *mapper* auch bei der Verteilung der *reads* zu vergleichbaren Ergebnissen führen. Die Laufzeit der *mapper* unterscheidet sich jedoch stark aufgrund des vergleichsweise langsamen BLAST Verfahrens von RNAseqMapper gegenüber der BWT(siehe Abb. 6).

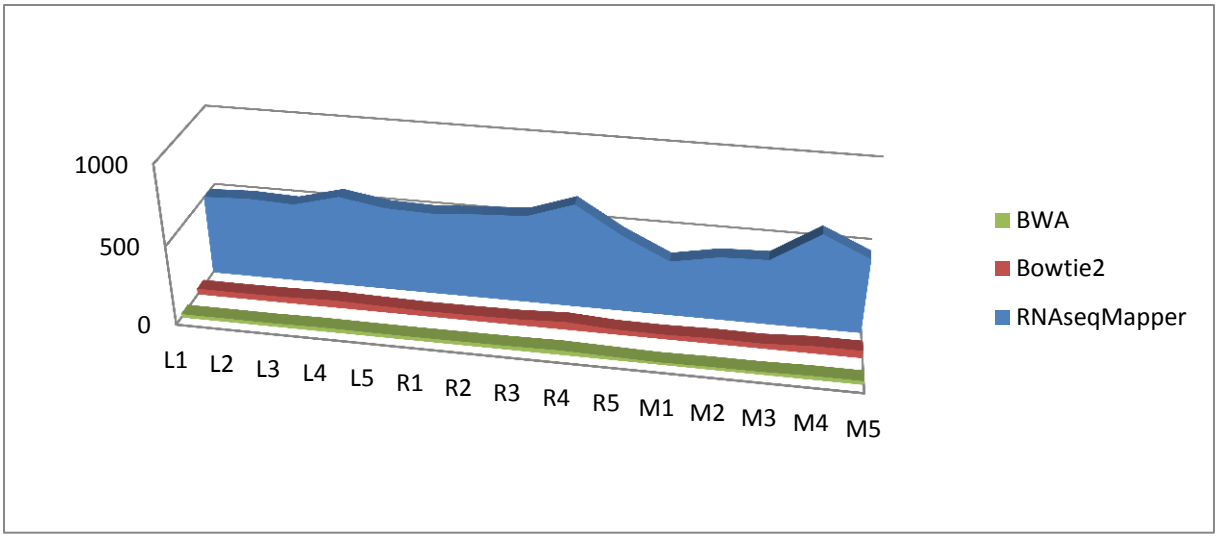

Abb. 6: Laufzeit der *mapper* beim *mapping* von RNA-Seq Daten auf *B. licheniformis* DSM13 in Minuten
Die Y-Achse gibt die Zeit in Minuten an, die X-Achse listet die Datensätze auf

Anhand von Abb. 6 zeigt sich der Vorteil des BWT basierten *mappings* gegenüber der BLAST-Methode. Während der RNAseqMapper für einen Datensatz ca. 5 Stunden benötigt, braucht Bowtie2 ca. 40 und BWA ca. 25 Minuten. Der Geschwindigkeitsvorteil sowie die geringfügig höhere Anzahl an *mapped reads* zeigen den Vorteil der BWT basierten *mapper*. Ein interessantes Phänomen tritt jedoch auf wenn man die Genomsequenz reversekomplementiert. RNAseqMapper und BWA erreichen die gleiche Anzahl an *mapped reads* auf der reversekomplementierten Sequenz wogegen bei Bowtie2 leichte Unterschiede im Bereich von ca. 100 *reads* pro Datensatz in der Anzahl der *mapped reads* auftreten. Worauf dieser Effekt basiert wurde nicht weiter untersucht.

Die TEX behandelten Datensätze wurden nur mit dem RNAseqMapper *mapped* und nicht im Vergleich behandelt. Tabelle 5 gibt eine Übersicht über die Verhältnisse bei diesen Datensätzen.

Tabelle 5: Übersicht über die Anzahl der *mapped reads* in den TEX behandelten Datensätzen

	Phase 1	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
Mapped	227.656	427.097	300.769	311.230	368.071
Unmapped	457.689	630.907	1.163.040	1.293.062	1.376.881
Multimapped	3.915.364	3.965.058	2.633.377	2.231.724	2.623.351

Die Laufzeiten des RNAseqMappers sind eindeutig länger, was auf den zeitaufwendigen BLAST zurückzuführen ist. Die auf dem Burrows-Wheeler-Algorithmus basierenden *Mapper* sind in dieser Hinsicht im Vorteil. Aufgrund dieser Tatsache und der vergleichbaren Anzahl an erfolgreich *gemappten reads* empfiehlt sich Bowtie2 als *Mapper* für Daten die den getesteten *Bacillus* Daten entsprechen. Es ist durchaus denkbar dass andere Daten (wie

z.B. eukaryotische RNA-Seq Daten) andere *mapper* als die hier verglichenen verlangen. Letztlich gibt es nicht den „idealen Allzweckmapper“, daher ist es wichtig Ergebnisse möglichst vieler *mapper* nutzen zu können. Zu diesem Zweck wurde das *tool* SAMtoTDS entwickelt, dass es erlaubt SAM formatierte *mappings* in ein TraV-kompatibles Format zu konvertieren.

5 Entwicklung eines Visualisierungs- und Analysetools für mehrere Transkriptomdatensätze

Transkriptomdatensätze sind stets spezifisch für die experimentellen Bedingungen, unter denen die Zellen angezogen wurden. Daher können Analysen, die auf diesen Datensätzen durchgeführt werden, auch nur solche Effekte (wie z.B. die Transkription von sRNAs) finden, die spezifisch für die herrschenden Bedingungen sind. Um die Menge der identifizierbaren Effekte zu erweitern, ist es notwendig, die Informationen aus verschiedenen Datensätzen kombinieren und vergleichen zu können. Damit können mehr Effekte, die spezifisch nur in einzelnen Bedingungen auftreten, gefunden werden und gleichzeitig ist es möglich, diese Effekte in den Kontext der Bedingungen zu stellen.

Die Annahme ist, dass bestimmte regulatorische RNAs spezifische Antworten der Organismen für bestimmte Bedingungen darstellen und daher eine möglichst gemeinsame Analyse aller Datensätze ein genaueres Bild der Regulation und der Gesamtausstattung an regulatorischen RNAs ergibt. Die zu Anfang dieser Arbeit verfügbaren *tools* verwendeten die in den SAMs/BAMs gespeicherten Informationen für ihre Visualisierung und Analysen. Durch die Verwendung dieser Einzelreadinformationen ergeben sich ein hoher Speicherverbrauch und lange Ladezeiten für die jeweiligen SAMs/BAMs. Der Speicherbedarf ist hierbei proportional zur *coverage* des Sequenzierexperiments. Mit der Entwicklung der Sequenziertechnologien hin zu immer größeren Readzahlen pro Experiment werden die Speicheranforderung für diese Programme noch steigen (Thürmer, 2014). Diese Situation zeigte den Bedarf nach einem neuen, analytischen *tool* mit sehr effizientem Speichermanagement. Die Methode von Wurtzel et al. (Wurtzel *et al.*, 2010), verwendet einen Ansatz, bei dem der Speicherbedarf mit der Basenanzahl des Genoms und nicht mit der Readanzahl korreliert was für das Ziel des Vergleichs vieler Datensätze miteinander vorteilhaft ist.

5.1 Design von TraV

In dieser Arbeit wurde mit TraV ein Analyseprogramm entwickelt, dass die Wurtzel et al. Methode implementiert und damit die simultane Analyse von vielen Transkriptomdatensätzen ermöglicht. Dabei wird bewusst die Einzelreadinformation von *mappings* verworfen um die dafür notwendige Speichereffizienz zu erreichen. Dies hat Einfluss auf das Anwendungsgebiet von TraV. Im Gegensatz zu anderen Analysetools, welche die Einzelreadinformationen behalten, ist TraV nicht in der Lage *single nucleotide polymorphism* (SNP) Analysen durchzuführen. Stattdessen wurde TraV in Design und Implementation auf die Analyse und Identifikation von Bereichen der transkriptionellen Aktivität ausgelegt. Damit liegt der Fokus der analytischen Methoden auf der Identifizierung von regulatorischen RNA-

features des Genoms, die über ihre spezifische transkriptionelle Aktivität (siehe Kapitel 6 für Beispiele solcher *features*) entdeckt werden können. Desweiteren kann TraV, wie vergleichbare *tools*, die transkriptionelle Aktivität von bereits annotierten *features* oder benutzerdefinierten *features* in Form von NPKMs berechnen.

TraV gliedert sich in zwei Komponenten, i) die TraV Analysesoftware welche in Java implementiert ist und ii) den SAM-Konverter SAMtoTDS, welcher Aufgrund der Speichereffizienz in C++ implementiert wurde. Der Konvertierungsschritt vom SAM zum TraV Austauschformat TDS ist im TraV Arbeitsfluss der speicheraufwendigste Teil. Da die Konvertierung zum TDS aber nur einmalig und außerhalb der TraV Analysesoftware durchgeführt wird, beeinflusst dieser Schritt den Speicherbedarf des Analyseteils von TraV nicht. Die TraV Analysesoftware ist als *webtool* implementiert und ermöglicht damit TraV auf einem dedizierten Server einzusetzen. Dies erlaubt die Arbeit mit TraV an Desktop PCs welche nicht über die notwendige Ausstattung zum Arbeiten mit Transkriptionsdatensätzen verfügen. Die serverseitige Implementierung ermöglicht darüber hinaus Arbeitsgruppen ihre Daten zentral zu verwalten und stellt damit die Synchronisation der Datensätze sicher und verringert den Aufwand beim Austausch der Mappinginformationen zwischen Kooperationspartnern. Um die Datensicherheit zu gewährleisten verfügt TraV über ein Usermanagement mit verschiedenen Rollen. Der Serveradministrator verfügt über alle Rechte und ist die einzige Person, die berechtigt ist, neue Projekte anzulegen sowie die Zugangsrechte von Benutzern auf die einzelnen Projekte zu bestimmen. Projektadministratoren verwalten die verfügbaren Transkriptomdatensätze ihrer Projekte und besitzen somit das Recht, alte Datensätze zu exportieren oder zu löschen und neue hinzuzuladen. „Einfache Benutzer“ besitzen nur Leserechte für die ihnen freigegebenen Projekte und dürfen die Datensätze zwar im vollen Umfang analysieren aber nicht verändern.

Für die Implementation der Benutzeroberfläche wurden Java Server Pages (JSP) verwendet. JSP ist eine Implementation der Java Programmiersprache welche auf Webservern, sogenannte *container*, eingesetzt wird. JSP liegt dabei als *layer* unterhalb des HTML *layers* vor und kann von diesem aus angesprochen werden. So werden zum Beispiel Benutzereingaben an den Java *container* weitergereicht, wo dann innerhalb einer Java Umgebung diese Eingaben verarbeitet werden können. Dies erweitert den Webserver in seinen Möglichkeiten über den in HTML und Javascript üblichen Funktionsumfang und erlaubt damit die Verarbeitung von komplexen Aufgaben auf dem Server. Die Wiedergabe von Ergebnissen geschieht dabei stets in Form von HTML, das innerhalb der Java Umgebung vorbereitet wird und an den Webserver zurückgegeben wird, welcher es dann in einer HTML-Seite eingebettet dem Benutzer darstellt.

Für die Darstellung von Grafiken in TraV werden *Scaleable Vector Graphics* (SVG) verwendet. SVG ist eine XML Sprache ähnlich HTML, die aber auf die Generierung von hochwertigen, interaktiven Grafiken ausgelegt ist. SVGs haben den Vorteil, dass sie als Skript generiert werden können und Vectorgrafiken sind. Das heißt, sie skalieren dynamisch und sind somit unabhängig von der Auflösung, in der sie dargestellt werden. Zusätzlich ist es möglich interaktive Komponenten in die Grafiken einzubauen welche mittels JavaScript gesteuert werden können. Damit ist es möglich sehr komplexe *Interfaces* zu implementieren, welche dank der JSP Umgebung über einen hohen Funktionsumfang verfügen können.

Für die Speicherung aller relevanten Daten verwendet TraV PostgreSQL. Dies erlaubt es TraV die zu verarbeitenden Daten schnell und effizient zu verwalten. Genauere Informationen zur Datenbankstruktur stehen im Kapitel 5.4.

5.2 Speicherbedarf der verschiedenen Methoden

Abb. 7 zeigt den geschätzten Speicherverbrauch der TraV Methode sowie verschiedener Ansätze zur Verarbeitung von *reads*. Hierbei wurde für die Berechnung des Speicherbedarfs der TraV-Methode die genomische Größe von *Bacillus licheniformis* DSM13 verwendet.

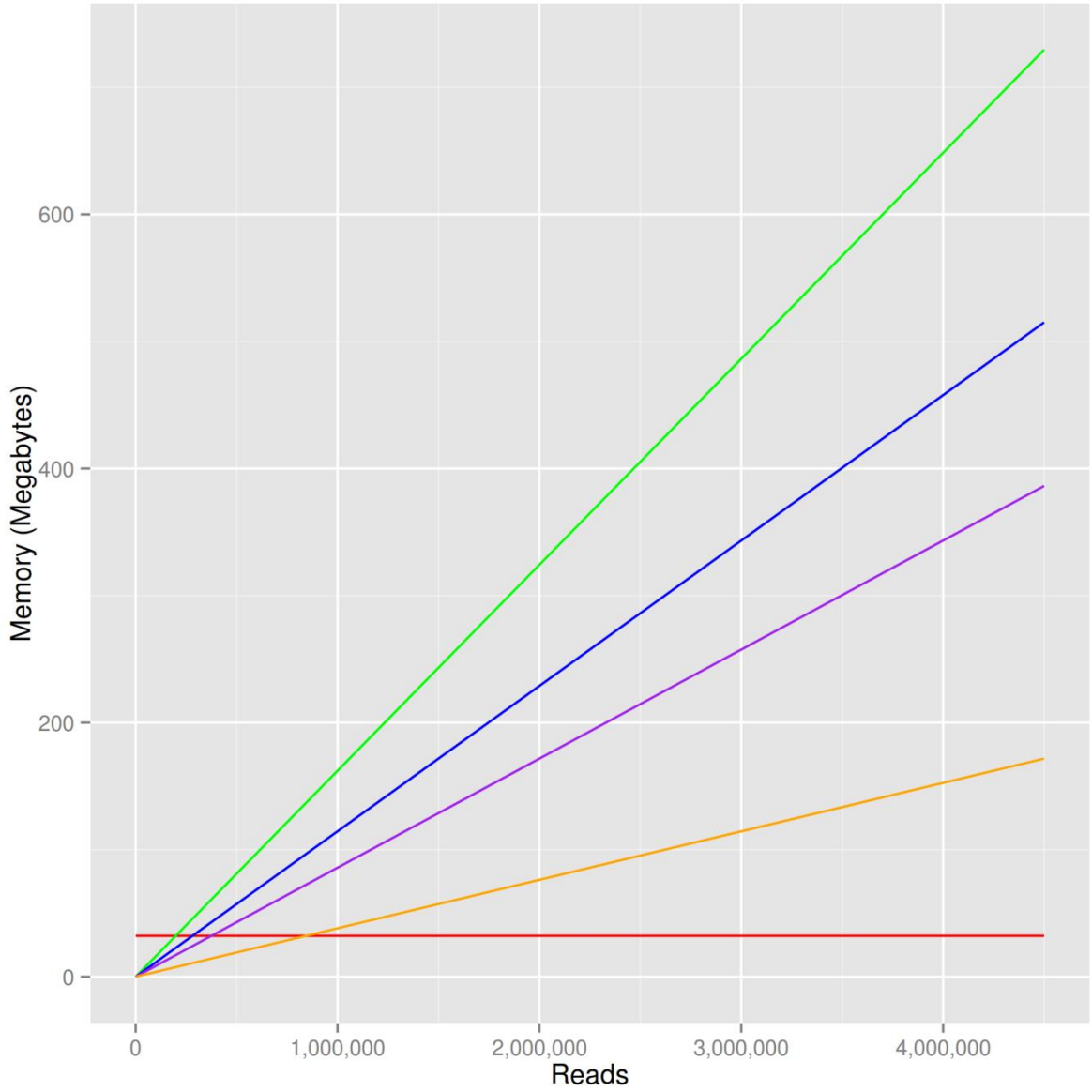

Abb. 7: Theoretischer Speicherbedarf der verschiedenen Methoden zur Handhabung von RNA-Seq Datensätzen
Gezeigt ist der Speicherbedarf in Megabyte in Abhängigkeit von der Anzahl der *mapped reads*. Alle visualisierten Daten entstammen den in dieser Arbeit beschriebenen Transkriptomdatensätzen. Der rote Graph zeigt den Speicherverbrauch der TraV Methode. Der grüne Graph zeigt den Speicherverbrauch wenn für *reads* neben den Koordinaten auch die Sequenz und Name mitgespeichert werden. Der blaue Graph zeigt den Speicherverbrauch wenn Koordinaten sowie Name gespeichert werden. Der lila Graph zeigt den Speicherverbrauch wenn neben den Koordinaten nur Sequenz gespeichert wird. Der orange Graph zeigt den Speicherverbrauch wenn ausschließlich die Koordinaten für jeden *read* gespeichert werden

Für *reads* wird die in den Sequenzierexperimenten verwendete Readlänge von 50 Basen benutzt. Der Speicherverbrauch der TraV-Methode ist allein abhängig von Genomgröße und

steigt daher nicht mit der Anzahl an verwendeten *reads*. Der Speicherverbrauch, bei einem Programm das Einzelreadinformationen speichert, hängt stark von der Implementation sowie den gespeicherten Informationen ab. Für die Speicherung eines strangspezifischen *reads* werden mindestens drei Werte benötigt, i) die Startkoordinate des *mappings*, ii) die Endkoordinate des *mappings* oder die Readlänge, iii) ein boolscher Wert für die Stranginformation. Will man Unterschiede in der Basensequenz zwischen *read* und Genom untersuchen, wird die Sequenz des *reads* benötigt. Zusätzlich kann auch noch der Name des *reads* gespeichert werden, wenn man die Identität eines *reads* benötigt. In den Schätzungen für den Speicherverbraucht wurde für einen Integer Zahlenwert eine Größe von 4 Byte veranschlagt. Ein boolscher Wert benötigt 1 Bit. Zeichenketten brauchen pro Zeichen 1 Byte. Zu diesen Größen wird jeweils ein Aufschlag für Objektstrukturen addiert, der sich aus dem notwendigen Speicheraufwand für den Aufbau eines Objekts in einer objektorientierten Programmiersprache ergibt. Die Größe dieses Aufschlags wurde Anhand der Größe einer minimalen Java Klasse auf 81 Byte geschätzt. Für die geschätzten Werte in Abb. 7. wurden alle Zahlenwerte als elementare Integers behandelt. Sollte eine Implementation Integer Werte mit Integer Klassen darstellen, fällt der Speicherbedarf entsprechend höher aus. Da dies in Anbetracht der Menge an zu verarbeitenden *reads* äußerst ineffizient wäre, wurde dieser Fall nicht genauer betrachtet.

Um die hohen Anforderung an Speicher für das Verarbeiten von einzelnen *reads* zu reduzieren, können verschiedene Methoden benutzt werden, die den Speicherverbrauch auf Kosten von Rechenzeit reduzieren. Da bei Visualisierungsprogrammen in der Regel immer nur ein kleiner Ausschnitt, genannt Fenster, des gesamten Genoms gezeigt wird, benötigt man auch nur die *reads,* die innerhalb dieses Fensters liegen. Mit den Start- und Stoppkoordinaten des Fensters ist es daher möglich zu bestimmen, welche *reads* aus einem SAM/BAM geladen werden müssen. Zu diesem Zweck werden die *reads* in einem SAM/BAM indexiert und geordnet, um so das Auffinden der benötigten *reads* zu beschleunigen. Der Nachteil dieser Methode ist, dass bei einer Veränderung des Fensters sämtliche *reads* erneut durchsucht werden müssen. Der Abgleich der Fensterskoordinaten mit Readkoordinaten verlangt dann jedes Mal Rechenzeit. Um diesen Vorgang zu beschleunigen, kann man das Genom in Abschnitte einteilen und anhand dieser Abschnitte die Zeilen des BAM/SAM zuordnen, innerhalb derer die *reads* für den jeweiligen Abschnitt liegen. Statt bei der Veränderung des Fensters die einzelnen *reads* zu überprüfen, können die Abschnitte überprüft werden, ob sie innerhalb des Fensters liegen. Dieses Verfahren verlangt jedoch eine Vorprozessierung um die *reads* den Abschnitten zuzuteilen. Beiden Verfahren ist gemein, dass mit zunehmender Fenstergröße der Speicherbedarf steigt, da mit zunehmender Fenstergröße auch die Anzahl der zu ladenden *reads* steigt.

Betrachtet man die Entwicklung innerhalb der Sequenziertechnologien, zeigt sich ein Trend zu immer größeren Leselängen und größerer Menge an *reads* (Thürmer, 2014). Programme, die bei Einzelreads die Sequenzinformation mitspeichern, werden mit der steigenden Leselänge auch mehr Arbeitspeicher benötigen um diese Informationen zu prozessieren. TraV hat den Vorteil, dass es von der Anzahl der *reads* unabhängig ist und der Speicherbedarf nur mit der Genomgröße korreliert.

5.3 Trav-Interface

Das Entwicklungsziel von TraV, nämlich die gleichzeitige Analyse von mehreren Transkriptomdatensätzen, verlangt die Möglichkeit, diese parallel in der Arbeitsoberfläche darzustellen und zu bearbeiten. Aus diesem Grund wurde TraV darauf ausgelegt, alle angebotenen Funktionen automatisch auf mehrere Datensätze anwenden zu können. Dies betrifft einfache Navigation der Datensätze bis hin zu komplexen Arbeitsschritten wie den analytischen Methoden. Das Webinterface erlaubt dank der HTML Implementierung problemlose Darstellung von beliebig vielen Datensätzen, wobei TraV derzeit auf 20 Datensätze pro Benutzer begrenzt ist. Um die Darstellung von vielen Datensätzen zu vereinfachen, bietet TraV die Möglichkeit, alle geladenen Datensätze in einem *multiline graph* zu vereinen, welcher für jeden Datensatz eine Kurve in nur einer Grafik darstellt und damit die Übersichtlichkeit der Datensätze verbessert. Abb. 8 zeigt die TraV Oberfläche.

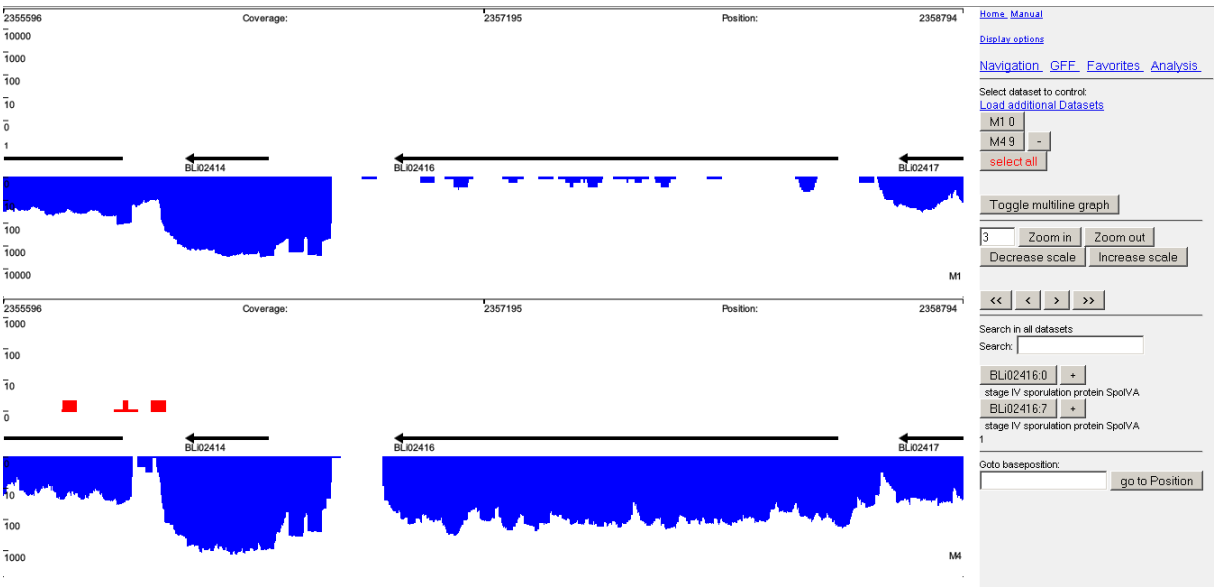

Abb. 8: Übersicht der TraV Benutzoberfläche
In der Mitte des Bildschirms werden die Graphen der geladenen Datensätze angezeigt Am rechten Rand ist das Menü für die Benutzereingaben. Das Menü selber hat mehrere Seiten, welche über die hier blauen *links* am oberen Rand des Menüs ausgewählt werden. Neben der Navigation wird in diesem Menü bestimmt, mit welchen Datensätzen interagiert wird. Neben der Navigation mit Basenpositionen ist auch eine Suche und Navigation mit Annotationen des Genoms möglich

In Abb. 9 wird die Funktionsweise der *multiline* Darstellung gezeigt. Diese Darstellungsform bietet sich bei vielen Datensätzen an um die Übersichtlichkeit zu erhöhen.

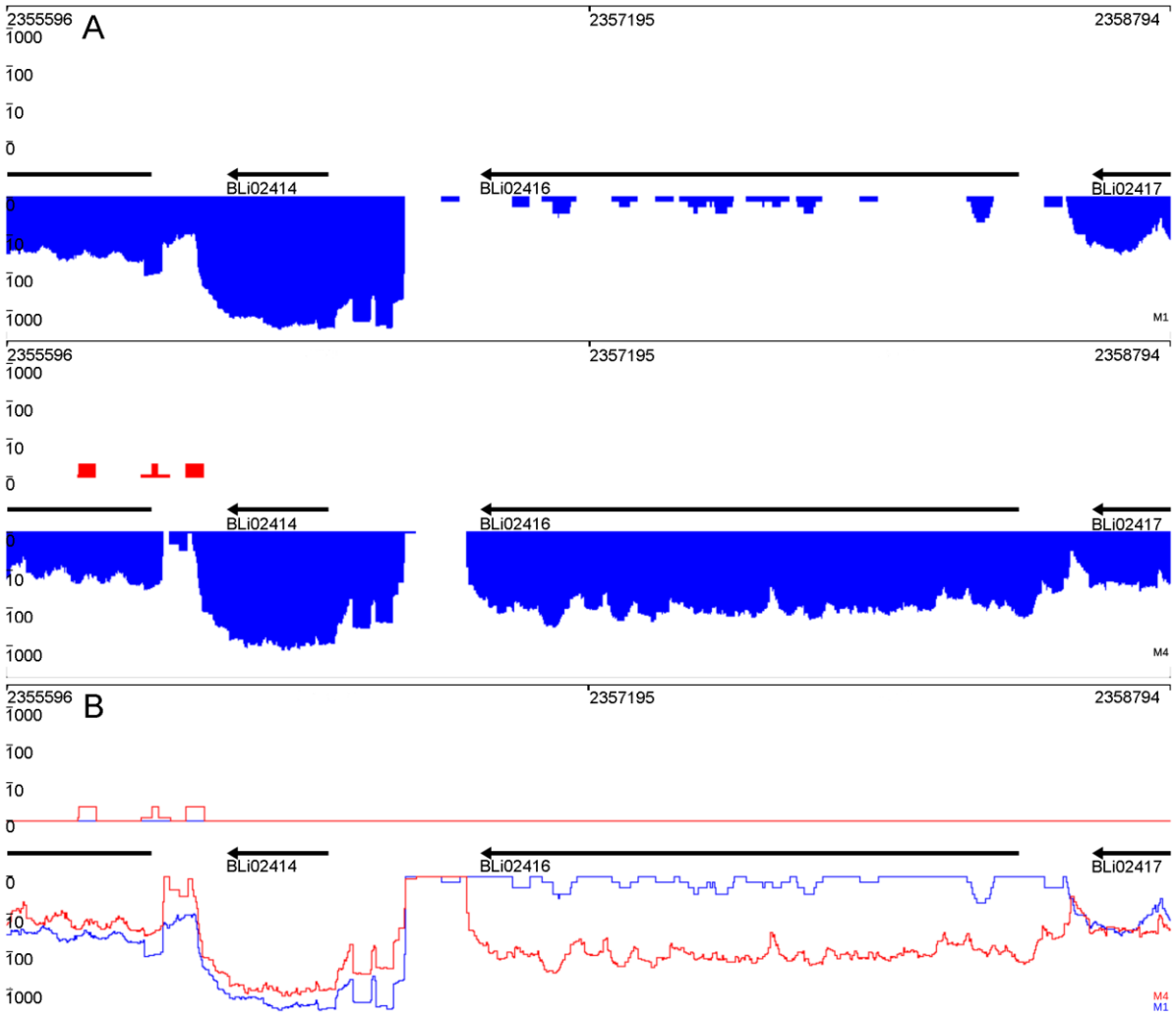

Abb. 9: Benutzeroberfläche von TraV mit Einzel- und Multigraph-Darstellung
Grafik A zeigt zwei RNA-Seq Datensätze (M1 und M4). Grafik B zeigt den *multiline* Graph für die in A dargestellten Datensätze. Alle hier darstellten Graphen sind logarithmisch skaliert

Die Grafiken innerhalb von TraV sind interaktiv und erlauben den Zugriff auf Positionsinformationen und Basenaktivitäten der Datensätze. Dies erlaubt ein direktes Ablesen und Analysieren der Transkriptionsaktivitäten durch den Benutzer in intuitiver Form. Die Interaktivität der Graphen ist ebenfalls auf die Verwendung von mehreren Datensätzen ausgelegt und synchronisiert die Positionseingaben zwischen den einzelnen Graphen. Abb. 10 zeigt ein Beispiel für diese Interaktivität. Auf diese Weise ist es möglich, Positionsinformationen und Basenaktivitäten auf mehrere Graphen abzufragen ohne dass eine separate Eingabe für die einzelnen Graphen notwendig wird. Alle Graphen in TraV sind aufgrund der großen Aktivitätsunterschiede die in Transkriptomen vorkommen können logarithmisch skaliert.

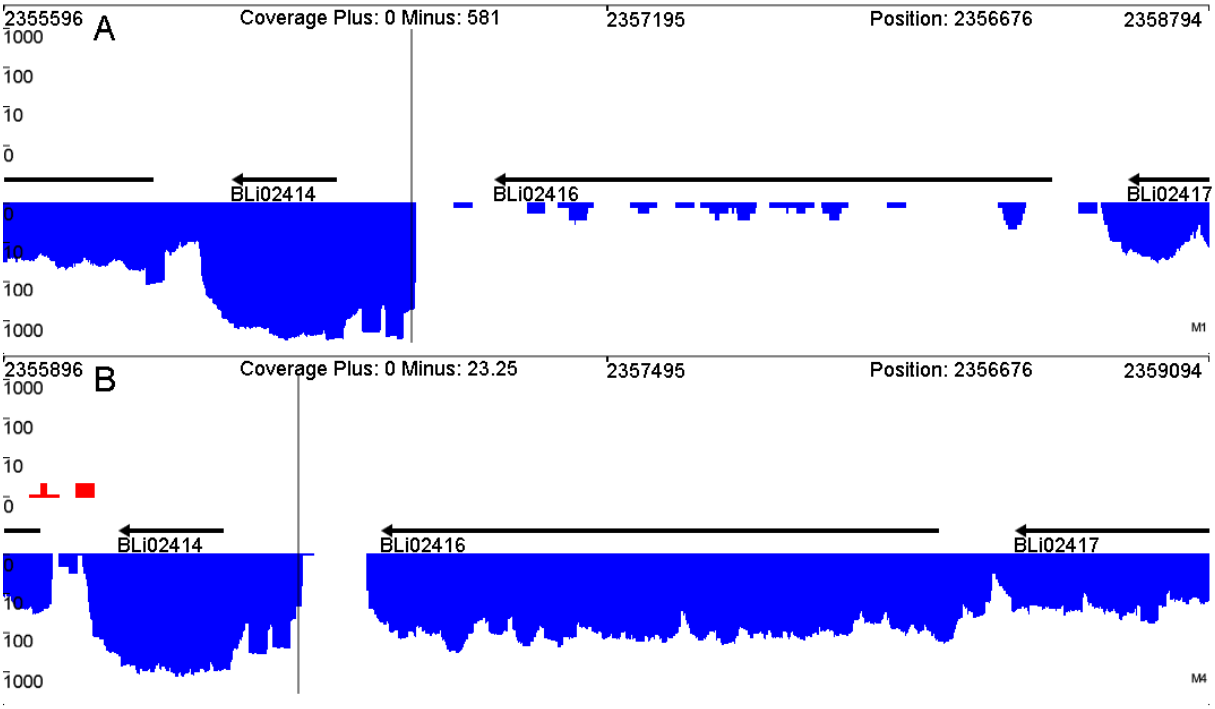

Abb. 10: Interaktivität innerhalb der TraV Graphen
Grafik A und B zeigen den gleichen Auschnitt vom Genom mit einer Versetzung um 300 Basen nach rechts in Grafik B. Der schwarzmarkierte Balken basiert auf der Interaktion mit dem Graphen und gibt die ausgewählte Position wieder, dessen Koordinate und Basenaktivität in der Kopfzeile des Graphen angezeigt wird. Interaktion mit einem Graphen resultiert in automatischer Übertragung der Interaktion auf alle Graphen so dass Basenaktvititäten und Positionen zwischen den Graphen direkt verglichen werden können

Die Navigation der Datensätze in TraV erlaubt eine Vielzahl von verschiedenen Herangehensweisen. Da je nach wissenschaftlicher Fragestellung z.B. bestimmte Gene oder aber ganze Bereiche mit mehreren Genen von Interesse sein können, erlaubt TraV sowohl genorientierte sowie positionsorientierte Navigation. Für die Navigation mittels Genen bietet TraV eine Suchfunktion, welche anhand von *locus tags* wie auch annotierten Produkten nach Genen sucht. Dabei gefundene Gene können dann einfach zur Navigation innerhalb der Datensätze verwendet werden. Diese Navigation bietet sich an wenn man z.B. das Expressionsverhalten eines oder mehrerer Gene zwischen verschiedenen Bedingungen vergleichen will. Positionsbasierte Navigation kann entweder direkt über die Eingabe einer Zielposition geschehen oder in relativen Schritten von der momentan dargestellten Position stattfinden. Dies ist vor allem dort interessant, wo nicht annotierte Bereiche im Genom untersucht werden sollen.

Neben den bereits im Genom vorhandenen Annotationen können vom Benutzer auch GFF3 formatierte Annotationen hinzugefügt werden. Abb. 11 zeigt ein Beispiel für die Darstellung dieser GFF Informationen in TraV. Diese benutzerdefinierten Annotationen erlauben die gezielte Navigation und sind in den analytischen Funktionen verwendbar. GFF3 *features*

ermöglichen die volle Plastizität der Annotationen und erlauben Ergänzung wie auch komplettes Ersetzen der genomspezifischen Annotationen innerhalb der analytischen Methoden.

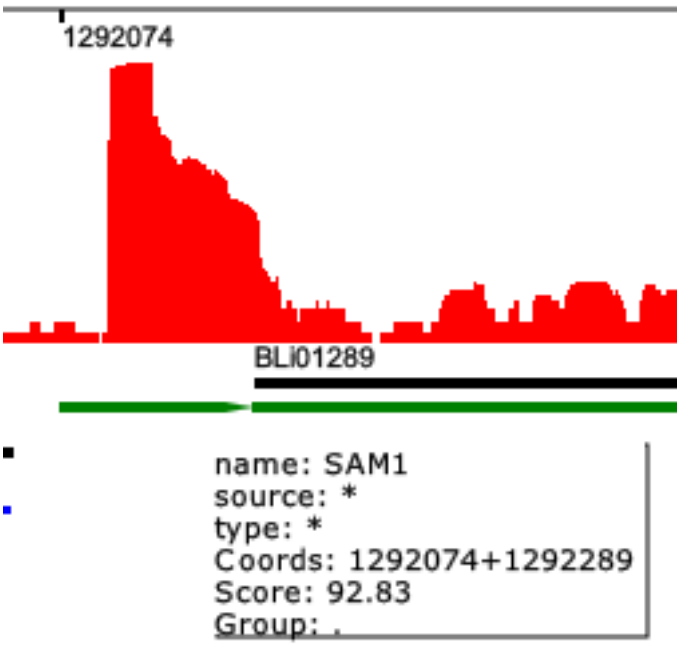

Abb. 11: Interaktion mit GFF Informationen in TraV Graphen
Die grünen Pfeile stellen GFF Informationen dar. Bei Interaktion mit diesen werden die GFF spezifischen Informationen eingeblendet. Diese Informationen können frei bestimmt werden solange sie das General Feature Format einhalten

Um innerhalb TraV das Untersuchen der Sequenz zu ermöglichen, bietet TraV den *magnification view*, eine gesonderte Darstellung bei der die dem *mapping* zugrundeliegende Genomsequenz mit den *coverage* Werten gezeigt wird. Die *magnification view* Funktion wird über Interaktion mit den Graphen in TraV aufgerufen und ist immer zentriert auf eine Position. Die Darstellung ist interaktiv und erlaubt das Vermessen der Abstände von Basenpositionen zu einer bestimmten Position. Dies dient der Suche nach *patterns*, wie z.B. Promotoren, welche in bestimmten Abständen zu *features* liegen. Abb. 12 zeigt beispielhaft, wie diese Darstellungsform von TraV benutzt werden kann, um *patterns* in Abhängigkeit von Expressionsprofilen zu suchen.

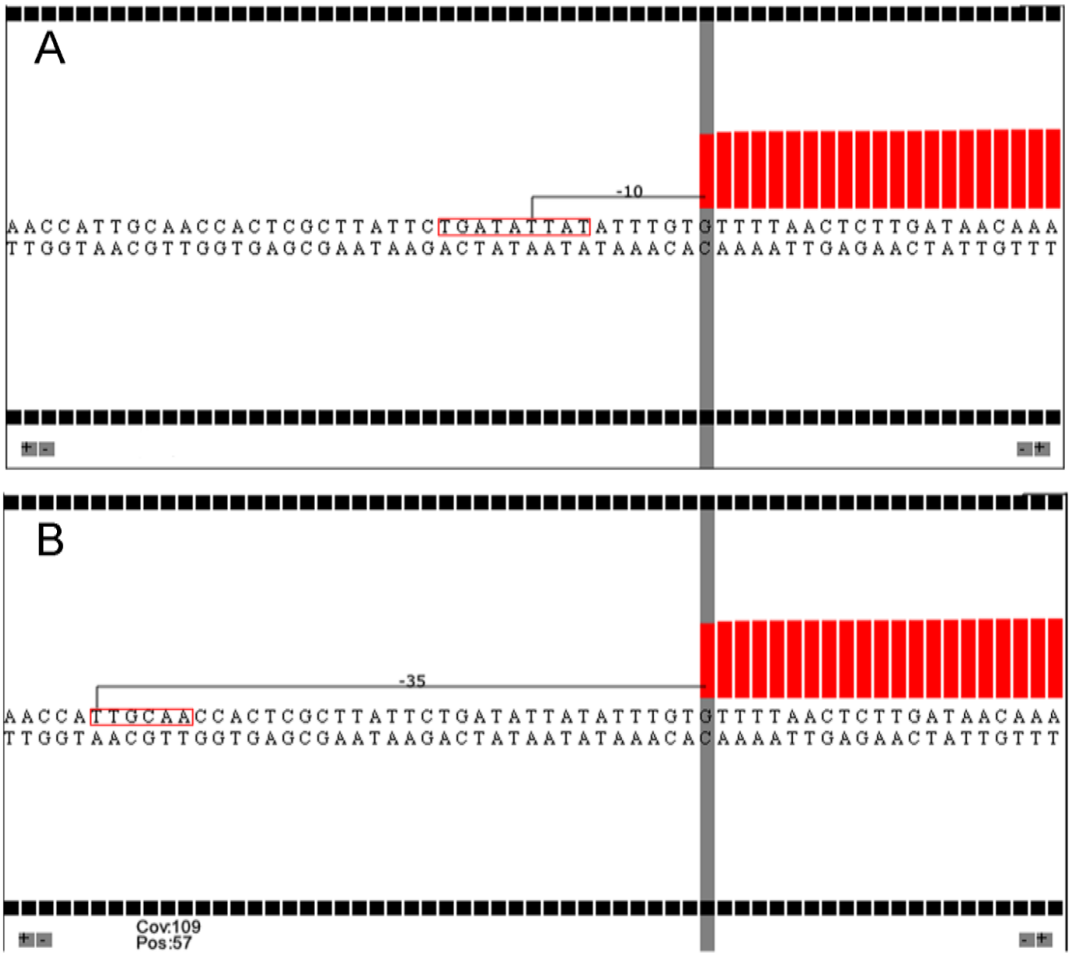

Abb. 12: TraV *Magnification View*
Diese TraV Darstellungsform erlaubt es die Sequenzinformationen mit den Basenaktivitäten detailiert zu korrelieren. Die Interaktivität des Graphen erlaubt es dieAbstände zwischen der markierten Base (grauer Balken) und anderen Positionen in der Genomsequenz zu vergleichen. In Grafik A beträgt diese Distanz 10 Basen *upstream* vom Ort des Transkriptionsstart (TSS). An dieser Position ist ein SigA -10 *pattern* erkennbar (TGaTATTAT). Grafik B zeigt die Position -35 vom TSS an, wo ein SigA -35 *pattern* erkennbar ist (TTGCAA)

5.4 Struktur der PostgreSQL Datenbank

Zur Speicherung der in TraV verwendeten Daten wurde ein *Entity-Relationship-Model* (ER-Modell) erstellt. Das ER-Modell beschreibt die Informationen der einzelnen Entitäten (*Entity*) in der Datenbank sowie deren Beziehungen (*Relationship*). Diese Beziehungen beschreiben die Struktur, in der die Entitäten in der Realität geordnet sind. Jeder Entität kann eine beliebige Anzahl an Attributen zugeordnet werden um diese genauer zu beschreiben. Zum Zugriff auf Daten werden in Datenbank-Modellen eindeutige Primärschlüssel (*primary keys*) verwendet. Innerhalb von TraV werden solche Primärschlüssel für die Genome, Replikons, Transkriptome und Benutzerkonten verwendet, da dies die primären Entitäten sind, über die auf die Daten zugegriffen wird. Für alle untergeordneten Daten werden keine Primärschlüssel erstellt. Dies spart Ressourcen, da auf eine eindeutige Indexierung verzichtet werden kann, da solche Daten stets als Block verarbeitet werden und spezifische Zugriffe auf einzelne Entitäten nicht nötig sind. In solchen Fällen werden sogenannte

Fremdschlüssel für den Zugriff verwendet, welche den Primärschlüsseln einer anderen Entität entsprechen. Als Beispiel können die Replikons dienen, welche als zentrale Ordnungsgröße in TraV dienen. Da in der Implementation von TraV Replikons stets als Gesamtes verwendet werden. Es ist daher nicht nötig für die einzelnen Gene eines Replikons eindeutige Schlüssel zu führen, da die Beziehung eines Gens zu einem Replikon ausreicht um dieses bei Verwendung aus der Datenbank zu referenzieren.

Innerhalb von TraV gibt es verschiedene Beziehungen, die vom ER-Modell beschrieben werden:

- Genome haben mehrere Replikons (1:N), jedes Replikon hat genau ein Genom, dem es zugeordnet ist
- Replikons haben mehrere Gene (1:N), Gene sind einem Replikon zugeordnet
- Genomische Sequenzen sind einem Replikon zugeordnet (1:1) und Replikons haben genau eine genomische Sequenz

Diese Beziehungen beschreiben die genomischen Informationen die TraV für die Darstellung und Auswertung verwendet. Diese Informationen werden mit Transkriptomdaten verbunden, welche über eigene Beziehungen verfügen:

- Transkriptome sind stets einem Replikon zugeordnet (1:1) womit sie automatisch auch einem Genom zugeordnet sind

Der Zugriff auf die Transkriptomdaten erfolgt über die zugeordneten Replikons. Um den Zugriff auf Genome und Transkriptome zu kontrollieren stellt TraV diese in Beziehung zu Benutzerkonten:

- Ein Benutzer kann Zugriff (Lese- und Schreibzugriff, wobei Schreibzugriff den Lesezugriff einschließt) auf mehrere Genome haben (N:M) womit auch der Zugriff auf die dem Genom zugeordneten Transkriptome und Replikons festgelegt wird, wobei jedes Replikon mehrere Transkriptome haben kann (N:M)

In TraV stellen Replikons die für die Datenorganisation maßgebende Entität dar während für die Verwaltung der Zugriffsrechte Genome als übergeordnete Entität verwendet werden. D.h. alle dem Genom zugehörigen Replikons und deren zugeordnete Daten können über das Genom kontrolliert werden während die Arbeit mit den Daten selber über die Replikon und Transkriptome Strukturen stattfindet. In Abb. 13 wird das ER-Modell grafisch dargestellt.

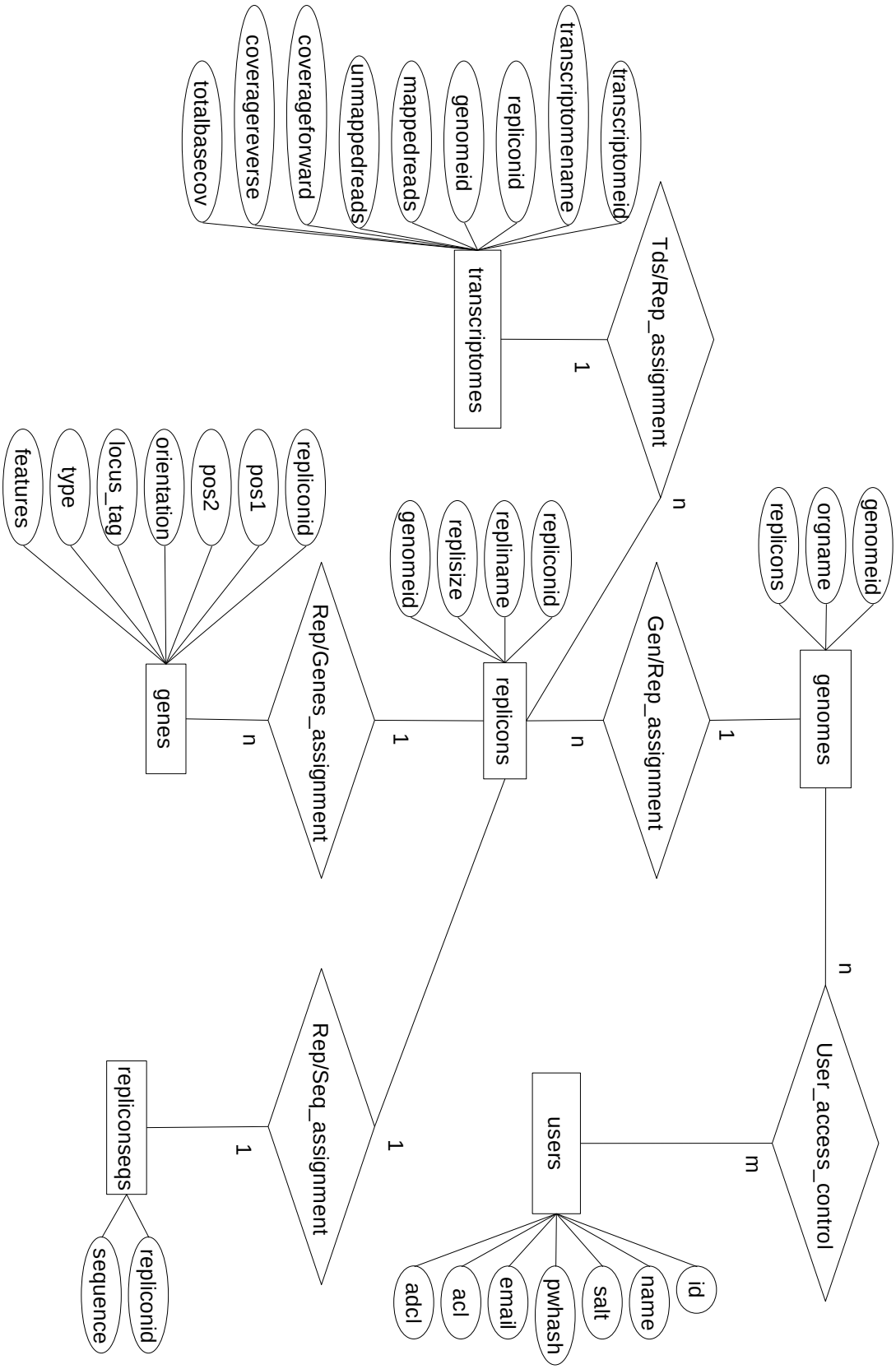

Abb. 13: ER-Modell der TraV-Datenbank
Rechtecke stellen die Entitäten dar, Ovale entsprechen den Attributen der Entitäten und Rauten repräsentieren die Beziehungen zwischen den Entitäten

Basierend auf dem ER-Modell wurde eine PostgreSQL Datenbank implementiert, wobei Genome, Transkriptome und Benutzerdaten jeweils eine dedizierte Datenbank erhalten. Jede Entität des Modells wird durch eine Tabelle innerhalb der entsprechenden Datenbank repräsentiert wobei jedes Attribut einer Spalte innerhalb dieser Tabelle entspricht. Tabelle 6 gibt eine Übersicht über die in der PostgreSQL Datenbank enthaltenen Tabellen. Tabellenamen sind fett markiert, Primärschlüssel sind unterstrichen, Fremdschlüssel sind doppelt unterstrichen.

Tabelle 6: Überblick über die Tabellen innerhalb der PostgreSQL Datenbank

Genomes	**Replicons**	**Genes**	**Repliconseqs**
genomeid	repliconid	Repliconid	Repliconid
orgname	repliname	pos1	sequence
replicons	replisize	pos2	
	genomeid	orientation	
		locus_tag	
		type	
		features	

Transcriptomes	**Users**
transcriptomeid	id
transcriptomename	name
repliconid	salt
genomeid	pwhash
mappedreads	email
unmappedreads	acl
coverageforward	adcl
coveragereverse	
totalbasecov	

5.5 Implementierung des Datenbankzugriffs

Damit die in 5.4 beschriebenen Datenbankstrukturen genutzt werden können, wird für TraV eine Java-Implementierung benötigt, die diese Strukturen schreiben und auslesen kann. Diese Implementierung sollte als eigenständige Klassen-Bibliothek stattfinden, sodass sie in anderen Tools neben TraV verwendet werden kann. Diese Klassen-Bibliothek wurde OmicsDatabase.jar genannt und stellt verschiedene Klassen zur Handhabung der PostgreSQL Datenbank und deren Entitäten sowie die Implementierung der in 5.6 beschriebenen Analysemethoden zur Verfügung.

Die Klasse DBManager dient der Handhabung der Verbindung zum PostgreSQL Server. Sie benutzt die Klasse ConfigManager welche die Zugangsdaten zum PostgreSQL Server verwaltet. Sie beinhaltet die grundlegenden Methoden über die SQL Befehle auf dem SQL Server ausgeführt werden. Zusätzlich stellt sie die Methoden zur Navigation und Verwaltung

der Datenbank zur Verfügung. Die Tabelle 7 gibt eine Übersicht über die Methoden der DBManager Klasse.

Tabelle 7: Beschreibung der DBManager Klasse und dessen Methoden

Methode	**Beschreibung**
createDB()	Erstellt eine neue Datenbank.
createMainDBs()	Erstellt die für TraV benötigte Datenbankstruktur. Sollte die Datenbankstruktur bereits bestehen, wird diese auf Vollständigkeit überprüft und gegebenenfalls repariert.
execPrepStatement()	Führt ein sogenanntes *prepared statement* aus. Dies ist ein SQL Befehl der zur Erstellungszeit vorbereitet wird und dadurch bei subsequenter Ausführung schneller ausgeführt werden kann als normale SQL Befehle.
execStatement()	Führt einen SQL Befehl aus und gibt das Ergebnis zurück falls vorhanden.
kill()	Beendet die Verbindung zum SQL Server.
prepareStatement()	Erzeugt ein *prepared statement*.
switchToDB()	Wechselt zur angebenen Datenbank auf dem SQL Server

Die Klasse DataSetHandler dient der Daten Ein- und Ausgabe mit der TraV Datenbank. Sie stellt eine Vielzahl an Funktionen zur Verfügung mit denen die einzelnen Datensätze wie z.B. Transkriptomdaten oder Genome verwaltet werden können. Sie benutzt zwei weitere Klassen, Auth und AuthHandler, welche für die Verwaltung der Benutzerzugangsrechte verwendet werden. Die Auth Klasse stellt den momentanen Benutzerzugang dar und wird von vielen Funktionen zur Zugangsüberprüfung (ein sogenanntes *token*) verwendet. Die AuthHandler Klasse generiert diese *tokens* und beherbergt auch die Verwaltungsmethoden für die Benutzerdatenbank. Innerhalb von TraV wird beim Anmelden eines Benutzers ein solches *token* generiert was für die Dauer der *Session* besteht.

Die Tabelle 8 gibt einen Überblick über die Methoden der DataSetHandler-Klasse.

Tabelle 8: Übersicht über die Methoden der DataSetHandler Klasse

Methode	**Beschreibung**
addGenome()	Fügt ein Genom in die Datenbank ein.
deleteGenome()	Entfernt ein Genom aus der Datenbank
getGenome()	Lädt ein Genome Objekt aus der Datenbank anhand der Genome Id
getGenomeByName()	Lädt ein Genome Objekt aus der Datenbank anhand des Genomnamens
getGenomeIDByRepID()	Gibt die Id eines Genoms basierend auf der Id eines zugeordneten Replikons.
getGenomeList()	Erstellt eine Liste aller verfügbaren Genome basierend auf den Zugangsrechten des Nutzers
getGenomeTrMap()	Erstellt einen Hash der für jedes zugangsberechtigte Genom die verfügbaren Transkriptome als Liste in Form von *key–value* Paaren enthält
renameGenome()	Benennt ein Genom um
purgeGenomes()	Löscht alle Genome (Nur Admin)
getReplikon()	Lädt ein Replikon als Replikon Objekt aus der Datenbank
getReplikonName()	Liefert den Namen eines Replikons basierend auf der Replikon ID
getReplikonNoBases()	Lädt ein Replikon als Replikon Objekt aus der Datenbank
getReplikonTrList()	Erstellt eine Liste der Transkriptom die für ein Replikon verfügbar sind
renameReplikon()	Benennt ein Replikon um
updateReplikon()	Ersetzt die Annotationen eines Replikons
getTranscriptome()	Lädt ein Transkriptom als TranscriptomeDataSet Objekt aus der Datenbank
importTranscriptome()	Importiert einen Transkriptomdatensatz in die Datenbank
deleteTranscriptome()	Entfernt einen Transkriptomdatensatz aus der Datenbank

getTranscriptomeMetaInfo()	Lädt Transkriptom-Metainformationen aus der Datenbank
getTranscriptomeOverview()	Erstellt einen Hash mit TranskriptomIDs als Schlüssel und den Metainformationen der entsprechenden Transkriptome als Werte
getTranscriptomesByGenome()	Generiert eine Liste an Transkriptomdatensätzen für ein Genom
getTranscriptomesByReplikon()	Generiert eine Liste an Transkriptomdatensätzen für ein Replikon

Für Genome, Replikons und Transcriptomdatensätze stehen entsprechende Klassen zur Verfügung, welche alle notwendigen Informationen beinhalten sowie verschiedene Funktionen bereitstellen, die der Verwendung dieser Daten dienen.

Für die Arbeiten mit genomischen Daten wurde die Replikon.java Klasse implementiert, welche sämtliche Informationen wie Genannotationen wie auch Sequenzdaten eines Replikons beherbergt. Zusätzlich bietet sie verschiedene Methoden an, welche das Verarbeiten der enthaltenen Informationen vereinfachen. Eine Übersicht über diese Funktionen gibt die Tabelle 9. Die Replikon Klasse verwendet zum Darstellen von Geninformationen die Gene.java Klasse, welche alle Informationen für ein Gen zusammenfasst.

Tabelle 9: Übersicht über die Methoden der Replikon Klasse

Methode	**Beschreibung**
createEmbl()	Erstellt eine EMBL formatierte Datei aus den im Replikon enthaltenen Informationen
getGeneByCoord()	Gibt ein Gene Objekt für eine Annotation an einer Position in Replikon zurück. Wenn an der angegebenen Position keine Annotation vorliegt, gibt die Funktion NULL zurück. Die „Lead“ und „Lag“ Varianten der Funktion sind strangspezifisch
getGeneByCoordLead()	
getGeneByCoordLag()	
getSeq()	Gibt die Sequenz aus einem Bereich des Replikons wieder
isGeneStartLead()	Diese Funktionen überprüfen, ob an einer Position im Replikon ein Gen beginnt oder endet. Diese Funktionen sind schneller in
isGeneStartLag()	
isGeneStopLead()	

isGeneStopLag()	der Ausführung als die getGeneByCoord() Funktionen
isInGene()	Überprüft, ob eine Koordinate innerhalb eines Genes liegt. Ähnlich zur getGeneByCoord() Funktion, gibt aber boolsche Werte zurück
isInGeneLead()	
isInGeneLag()	
readEmbl()	Funktionen zum Einlesen von EMBL und GenBank Daten. Normalerweise werden diese Funktionen nur vom Konstruktor der Klasse verwendet, welcher automatisch die passende Funktion benutzt
readGenBank()	
replaceGenes()	Ersetzt die Genannotationen im Replikon mittels eine Liste von featureBlock Objekten

Die TranscriptomDataSet.java Klasse ist das Kernstück von TraV. Innerhalb dieser Klasse werden die Basenaktivitätswerte verwaltet sowie eine Vielzahl von Methoden bereitgestellt. Diese Methoden dienen vor allem analytischen Zwecken, wie z.B. das Berechnen von NPKMs. Tabelle 10 gibt eine Übersicht über diese Methoden. Die im TranscriptomDataSet (TDS) enthaltenen Informationen sind stets spezifisch für ein Replikon und Informationen wie die Länge der Basenaktivitätslisten sind gleich der Länge des zugehörigen Replikons. Bestimmte Methoden verlangen zusätzlich Einzelreadinformationen, welche aber optional enthalten sind. Liegen keine Einzelreadinformationen vor, können diese Methoden nicht verwendet werden.

Tabelle 10: Übersicht über die Methoden der TranscriptomeDataSet Klasse

Methode	**Beschreibung**
calcNPKMvalue()	Berechnet einen NPKM Wert für einen Bereich des Replikons
calcRPKMvalue()	Berechnet einen RPKM Wert für einen Bereich des Replikons. Diese Methode verlangt dass Einzelreadinformationen im TDS vorhanden sind
calcReadsInRegion()	Berechnet die Anzahl an Reads die in einem Bereich gemappt wurden. Benötigt Einzelreadinformationen
getReadsInRegion()	Gibt eine Liste an Reads wieder, die in einem Bereich des mappen. Benötigt Einzelreadinformationen
cleanSinglets()	Entfernt Aktivitäten, welche auf einzelnen Reads ohne Überlapp mit anderen Reads basieren. Dient der Rauschfilterung
exportTDS()	Erstellt eine TDS Datei, welche von TraV importiert werden kann
getCov()	Gibt die Basenaktivität für eine Position wieder
merge()	Vereint das TDS mit einem anderen TDS

Einzelreadinformationen müssen bei der Erstellung des TDS explizit angefordert werden. Im normalen Ablauf verwendet TraV diese Informationen nicht. Sie zusätzlich mitzuführen führt dazu, dass TraV die in 5.2 beschriebene Speichereffizienz aufgeben muss. Daher sollten diese Informationen und die dazugehörigen Methoden nur dann verwendet werden, wenn sie z.B. zu Vergleichszwecken benötigt werden.

Diese vier beschriebenen Klassen sind die primären Arbeitsklassen für alle Vorgänge innerhalb von TraV. Neben diesen Hauptklassen gibt es verschiedene kleinere Klassen, die vor allem der Modellierung von Informationen dienen, wie z.B. GFF Einträge, Reads oder die featureBlock.java Klasse. Diese sekundären Klassen werden oft von den primären Klassen intern verwendet, können aber auch direkt für die Bearbeitung von spezifischen Fragestellungen benutzt werden.

5.6 Analysemethoden

Während der Arbeiten mit dem TraV-*Interface* zeigte sich, dass neben den im *Interface* möglichen Interaktionsmöglichkeiten mehr Methoden notwendig wurden, die die Analyse der Daten unterstützen. Diese Methoden sollten dabei so wenig Annahmen gegenüber den Intentionen der Analyse wie möglich machen, so dass die gefundenen Kandidaten als Grundlage für eine möglichst große Anzahl an analytischen Interessen dienen können und nicht nur auf spezielle Fragestellungen beschränkt sind. Aus diesem Grund wurde innerhalb der Implementation von TraV auf die Einbindung von externen Programmen wie z.B. das in 2.5.2 erwähnte Infernal verzichtet. Stattdessen wurden die analytischen Methoden auf die von TraV behandelten Informationen, nämlich transkriptionelle Aktivitäten und deren genetischen Kontext ausgerichtet. Deswegen werden von den Analysemethoden stets *loci* im Genom identifiziert, deren transkriptionelles Verhalten und Kontext stets definierte Voraussetzungen erfüllen. Als Folge dieser beschriebenen Voraussetzungen lassen sich keine festen Qualitätskriterien festlegen wonach diese bewertet werden könnten, da diese nicht *features* sondern lediglich Daten beschreiben. Damit werden von TraV nicht explizite *features*, wie z.B. ein *riboswitch* vorhergesagt, sondern *loci* deren transkriptionelle Eigenschaften mit den erwarteten *features* vereinbar sind. Alle TraV Vorhersagen sind somit Kandidatenlisten für nachfolgende analytische Methoden und werden auch entsprechend im Ergebnis aufbereitet, indem nämlich die *loci* mit deren Koordinaten als GFF3 formatierte Annotationen, NPKM Wertetabellen und Sequenzen für die Folgeanalysen bereitgestellt werden.

Die parallele Analyse mehrerer Datensätze findet über ein *merging* Verfahren statt, bei dem alle für die Analyse betrachteten Datensätze zu einem temporären Datensatz vereint werden. Auf diesem vereinten Datensatz werden dann die analystischen Methoden durchgeführt und so die Kandidatenlisten generiert. Anschließend werden dann in den ursprünglichen Datensätzen die Aktivitäten (NPKMs) der gefundenen Kandidaten berechnet. Durch dieses Verfahren können Kandidaten für *features* selbst in Datensätzen erkannt und verglichen werden, in denen das *feature* keine Aktivität aufweist und somit ohne das *merging* Verfahren nicht erkannt werden könnten.

Alle Methoden, mit Ausnahme der Suche nach *antisense* Transkripten und der Vorhersage von *transcriptional start sites* können jeweils strangspezifisch oder strangunspezifisch verwendet werden. Bei der strangunspezifischen Suche werden die Basenaktivitäten in den Transkriptomdatensätzen zu einem künstlichen Aktivitätsstrang aufaddiert und sämtliche Vorhersagen mit diesem ausgeführt, wobei die Stranginformation von Genen ignoriert wird. Dieses Vorgehen ist notwendig, wenn die RNA-Seq Daten nicht strangspezifisch erstellt wurden. Da so aber nicht mehr zwischen der Aktivität der einzelnen Stränge unterschieden

werden kann, ist die Vorhersagekraft bei solchen strangunspezifischen Datensätzen stark eingeschränkt.

5.6.1 Berechnung von NPKM Werten

NPKM steht für *nucleotide activites per kilobase of transcript per million mapped reads.* Sie sind normalisierte Vergleichswerte für die Expressionsstärke von *features* und dienen dem einfachen Vergleich von Expressionsstärken ohne statistische Methoden verwenden zu müssen. Da innerhalb der TraV-Methode die Einzelreadinformationen nicht zur Verfügung stehen, können RPKMs nicht berechnet werden ohne die Speichereffizienz der Methode aufzugeben. Um einen vergleichbaren Wert bereitzustellen, wurden NPKMs definiert (Wiegand *et al.*, 2013), welche vergleichbar zu RPKMs sind, aber zur Berechnung die Einzelbasenaktivitäten anstatt der Einzelreadinformationen benutzen und damit innerhalb der TraV-Methode anwendbar sind. Für NPKMs gelten die gleichen Beschränkungen wie für die in 2.4 beschriebenen RPKMs. NPKM Werte werden nach der folgenden Formel berechnet.

$$NPKM\ (n,m) = \ 10^9 \frac{\sum_{i=n}^{m} f(i)}{\sum_{i=1}^{l} g(i)(m-n)}$$

Die Variablen *n* und *m* stehen hier für die Start- und Stoppositionen des *features*, für das der NPKM Wert berechnet wird. Die Funktion *f(i)* gibt die Basenaktivität für einen bestimmten Strang an Position *i* wieder. Die Funktion *g(i)* errechnet die Summe der Basenaktivitäten beider Stränge an Position *i*. Die Variable *l* ist die Gesamtlänge des betrachteten Genoms.

TraV gibt die Möglichkeit für die *features* des Genoms wie auch benutzerdefinitierte *features* die NPKM-Aktivitätswerte zu berechnen. Bei dieser Berechnung können ein oder mehrere Transkriptomdatensätze verwendet werden. Das Ergebnis wird tabellarisch bereitgestellt und gibt jeweils den Namen des *features* sowie die NPKM-Werte für die benutzten Transkriptomdatensätze wieder, so dass diese entweder manuell oder maschinell direkt verglichen werden können.

In Abb. 14 ist die transkriptionelle Aktivität des *hag*-Gens aus den Phasen M1 bis M5 dargestellt. Die Tabelle 11 zeigt die dazugehörigen NPKM-Werte zum Vergleich.

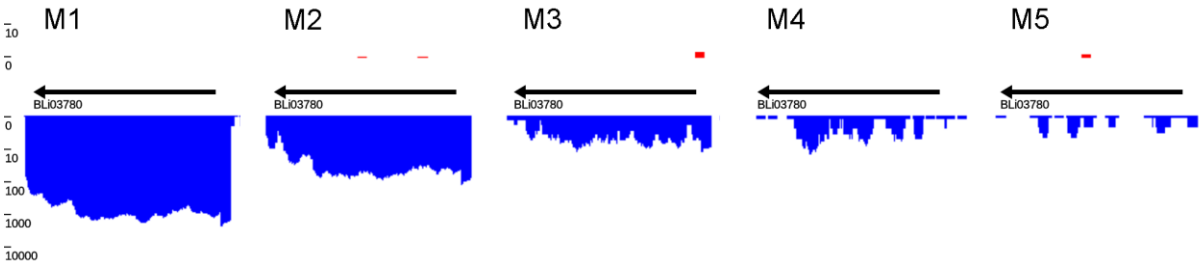

Abb. 14: Transkriptionelle Aktivität des *hag*-Gens (BLi03780) im Verlauf der Fermentation (Phasen M1 bis M5)
Blaue Graphen beschreiben die Aktivitätswerte der einzelnen Basen des Gens. Die Aktivität ist logarithmisch skaliert

Die unterschiedliche Expressionstärke des Gens in den einzelnen Phasen wird durch die NPKM-Werte bestätigt.

Tabelle 11: NPKM-Werte von *hag* in den Phasen M1 bis M5

Phase	M1	M2	M3	M4	M5
NPKM	6635	325	37	23	9

Der Vergleich über die NPKM-Werte erlaubt somit einen schnellen Vergleich der Genaktivität zwischen verschiedenen Transcriptomdatensätzen. Die Abschätzung einer differentiellen Expression ist jedoch oft schwierig, da nicht eindeutig gesagt werden kann, ab wieviel Differenz zwischen zwei NPKMs eine differentielle Expression vorliegt. Hierfür werden statistische Methoden benötigt. In Abb. 15 und Tabelle 12 ist ein Vergleich für das *degU*-Gen zu sehen, welches ohne statistische Absicherung nicht eindeutig als differentiell oder nicht-differentiell exprimiert identifiziert werden kann.

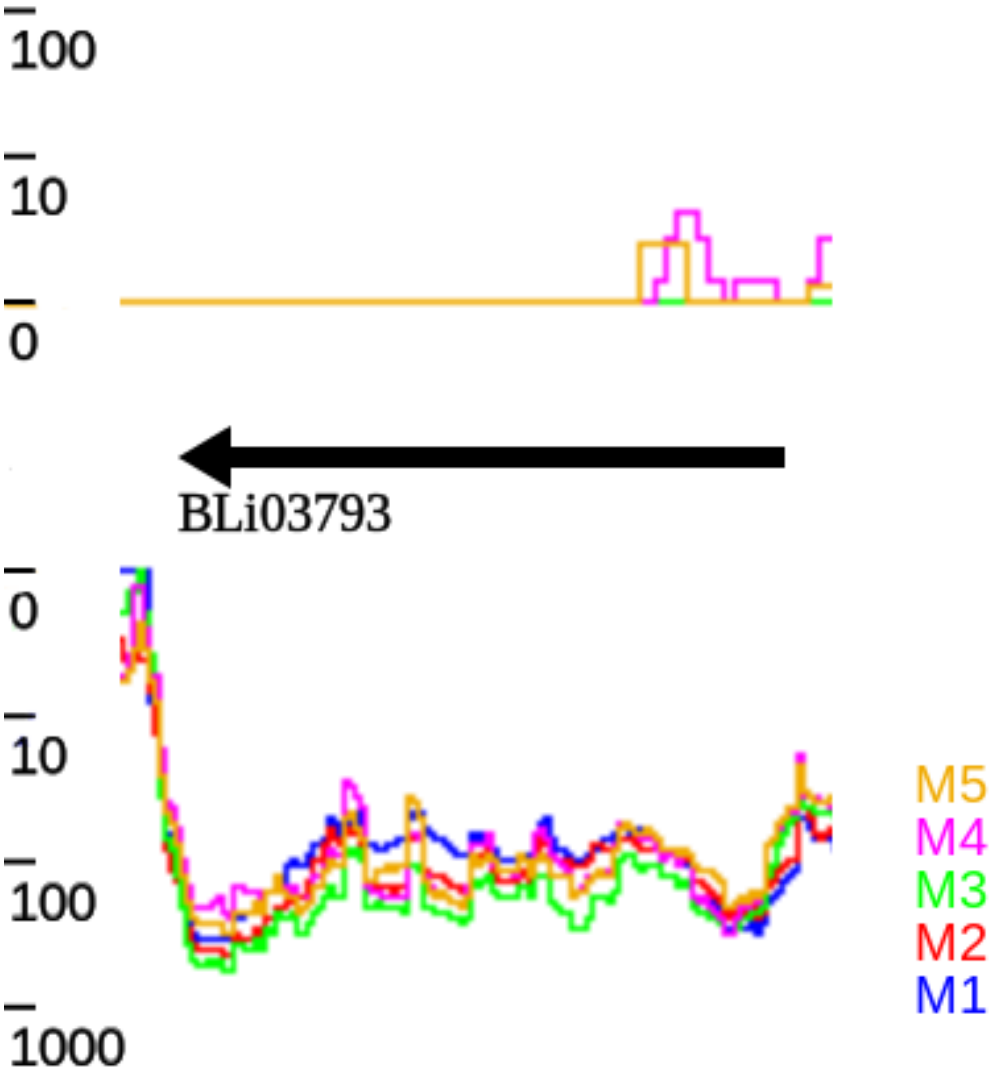

Abb. 15: *Multiline-graph* des *degU*-Gens (BLi03793) aus den Proben M1 bis M5
Die Aktivität in den verschiedenen Bedingungen liegt sehr nahe beieinander

Tabelle 12: NPKM-Werte von *degU* in den Phasen M1 bis M5

Probe	M1	M2	M3	M4	M5
NPKM	819	1009	1411	861	874

Das Expressionsverhalten von *degU* zeigt kein so eindeutiges differentielles Verhalten wie z.B. das *hag*-Gen.

5.6.2 Vorhersage von *transcriptional start sites* (TSS)

Die Startpunkte der Transkription (TSS) sind interessante Merkmale innerhalb des Genoms, da sie einen direkten Hinweis auf Promotoren und weitere in ihrer Umgebung zu erwartenden regulatorischen Signale darstellen (Busby and Ebright, 1994). Dank der genauen Auflösung der RNA-Seq Methode ist es möglich, TSS zu identifizieren die transkriptionell ausreichend aktiv waren. Solche TSS sollten sich über ein spezifisches

Muster, nämlich einen starken Anstieg der transkriptionellen Aktivität über einen Bereich von nur sehr wenigen Basen (Im Idealfall zwei Basen), identifizieren lassen. Eine große Herausforderung stellt dabei die zunehmende Schwankung der *coverage* bei großen Readmengen dar. In der Abb. 16 wird der Einfluss der *coverage* auf die Identifikation von Anstiegen in der Expression dargestellt.

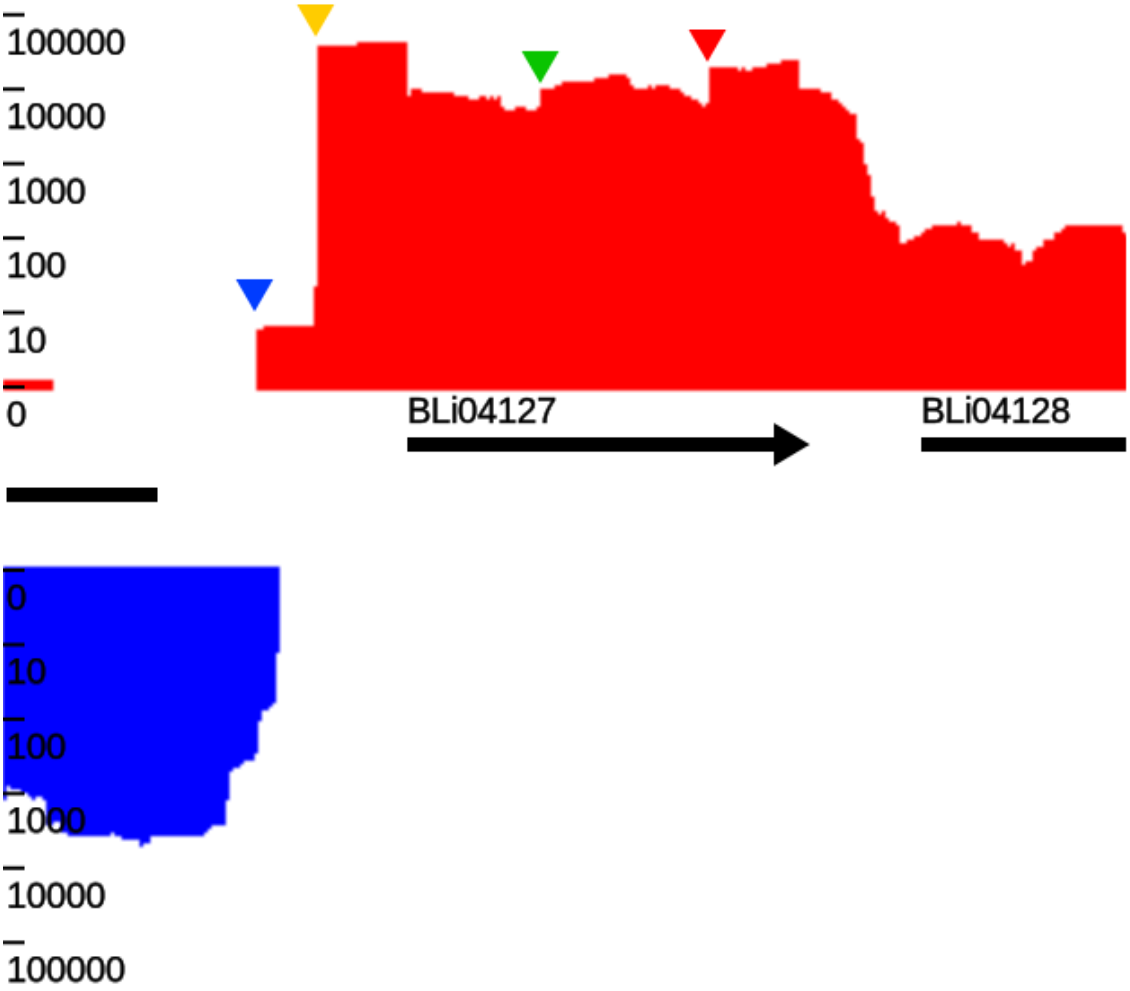

Abb. 16: Darstellung der Schwankung der c*overage* bei hoher Readanzahl
Der blau markierte Anstieg entspricht einer Differenz von 6 *reads*. Der gelb markierte Anstieg entspricht einer Differenz von ca. 40.000 *reads*. Der grün markierte Anstieg entspricht einer Differenz von ca. 4000 *reads.* Der rot markierte Anstieg entspricht einer Differenz von 13.000 *reads*. Die logarithmische Darstellung stellt die Größe dieser Anstiege ins Verhältnis zur umgebenden Aktivität

Anhand von Abb. 16 lassen sich vier interessante Anstiege zeigen. Der blau und der gelb markierte Anstieg sind zwei vielversprechende Kandidaten für TSS, da sie über einen erkennbar hohen Anstieg über kurze Distanz verfügen und am Anfang des aktiven Bereichs liegen. Der rot markierte Anstieg könnte ein Kandidat für eine TSS für das folgende Gen BLi04128 (*lanM2*) sein und wäre damit ein interner Promotor im Gen BLi04127(*lanA2*). Der grün markierte Anstieg ist im allgemeinen Vergleich der umgebenden Aktivitäten sehr gering und ist wahrscheinlich kein TSS. Aufgrund dieser c*overage* abhängigen Schwankungen wird bei der Suche nach TSS in TraV nicht die absolute Differenz der Anzahl der *reads* zwischen

zwei Basen für die Suche benutzt, sondern die logarithmischen Werte dieser Differenzen. Durch diese Vorgehensweise werden die *coverage* abhängigen Schwankungen abgeschwächt. Damit lässt sich ein Algorithmus definieren, der für die Suche nach TSS Kandidaten verwendet werden kann. Dazu wird an jeder Position des Genoms die Steigung über eine bestimmte Anzahl an Basen berechnet. Die Standarddistanz für das Steigungsdreieck ist 1. Wenn die TSS durch Degradationseffekte der mRNA uneindeutig werden, kann diese Distanz erhöht werden um dem Degradationseffekt entgegenzuwirken. Das hat aber zur Folge dass die Vorhersagen im Allgemeinen ungenauer werden. Neben dieser Distanz kann man einen *cut-off* definieren, unterhalb dessen die Anstiege nicht groß genug sind, um einen TSS-Kandidaten zu definieren. Innerhalb der *B. licheniformis* DSM13 Datensätze wurde ein *cut-off* von ln 4 verwendet, da dies dem kleinsten, experimentell bestätigten Anstieg einer TSS entsprach (Wiegand *et al.*, 2013). Dieser *cut-off* ist spezifisch für die jeweiligen Datensätze und muss dementsprechend empirisch bestimmt werden.

5.6.3 Suche nach 3‘ und 5‘ *untranslated regions* (UTR)

Innerhalb von mRNAs gibt es Bereiche, welche nicht über protein-kodierende Funktionen verfügen, in denen aber regulatorische *features* kodiert sein können. Diese Regionen werden *untranslated regions* (UTRs) genannt. Dabei sind die Enden von mRNAs, 3‘ und 5‘ UTRs von besonderem Interesse, da sich in diesen bekannte Regulatoren wie *riboswitches* und Terminatoren befinden können. Innerhalb von TraV wurden daher zwei Methoden zur Suche nach diesen UTRs implementiert. Der Algorithmus für die Suche nach 5‘ UTRs sucht dabei nach Regionen von transkriptioneller Aktivität die in Leserichtung an ein Gen grenzen und deren aktiver Bereich in Leserichtung außerhalb eines Genes beginnt. Dabei wird innerhalb des Basenaktivitätsgraphen nach Bereichen gesucht, welche mit einem Übergang von Null zu größer als Null Aktivität beginnen und anschließend auf ein Gen treffen ohne vorher wieder auf Null Aktivität zurückzufallen. Bei der Suche nach einer 3‘ UTR wird die Reihenfolge der Merkmale umgedreht. Gesucht wird nach Aktivität größer als Null, die in Leserichtung in einem Gen startet und dann außerhalb des Gens auf Null fällt, ohne vorher nochmals in Leserichtung auf ein Gen zu treffen. Die Abb. 17 zeigt zwei Beispiele für solche UTRs wobei Abbildung A eine 5‘ UTR und Abbildung B eine 3‘ UTR zeigt. Zusätzlich ist in Abbildung A ein grüner Pfeil gezeigt, welcher einen bekannten TPP-*riboswitch* vor dem Gen BLi03258 (*thiT*) markiert.

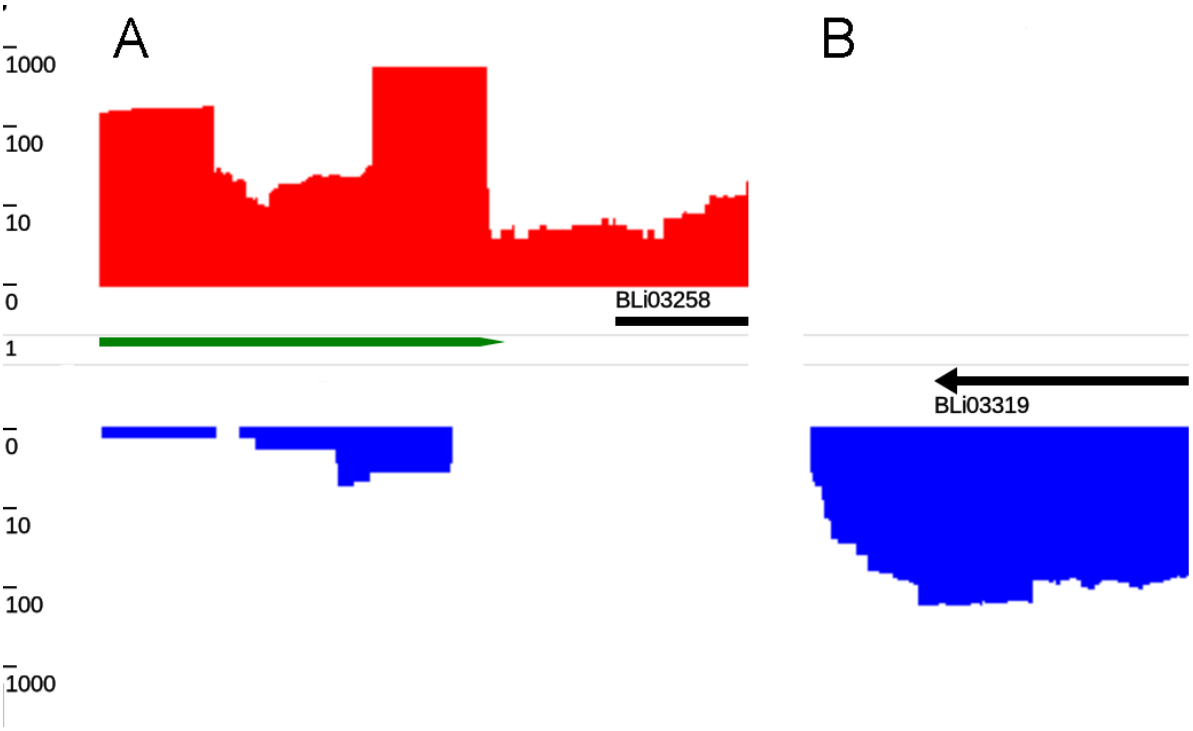

Abb. 17: Beispiele für UTRs
Abbildung A zeigt eine 5‘ UTR für das Gen BLi03258 (*thiT*). Abbildung B zeigt eine 3‘ UTR für das Gen BLi03319 (*yugI*). Der grüne Pfeil in Abbildung A markiert den bekannte TPP *riboswitch* des BLi03258 Gens

Für die von dieser Methode gefundenen Kandidaten wird zusätzlich die Länge der UTR und die NPKMs in den jeweiligen, für die Analyse verwendeten, Transkriptomdatensätze berechnet und angeben.

5.6.4 Suche nach Transkripten ohne zugeordnete Annotation (*Free transcipts*)

Wie in 2.1.1 erwähnt, gibt es regulatorische RNAs welche als distinkte Transkripte gebildet werden und somit in *trans* ihre Funktion innerhalb der Zelle auswirken. Die Annahme hierbei ist, dass diese regulatorischen RNAs innerhalb des Genoms in der Regel nicht annotiert sind. Sie bilden somit einen Bereich transkriptioneller Aktivität, der nicht mit einer Annotation überschneidet. Innerhalb von TraV werden diese Transkripte *free transcripts* genannt, da sie frei von einer Annotation sind. Der Algorithmus zur Suche nach diesen *free transcripts* sucht nach Bereichen transkriptioneller Aktivität, die außerhalb eines Genes beginnen und auch wieder enden ohne ein Gen in jedwede Leserichtung einzuschließen. Dies bedeutet dass auf einem Strang nach Bereichen im Aktivitätsgraphen gesucht wird, deren Aktivität von Null zu größer Null steigt und auch wieder auf Null zurückfällt ohne jemals ein Gen geschnitten zu haben. In Abb. 18 ist ein Beispiel für eine regulatorische RNA, *bsrG* (Jahn and Brantl, 2013) gezeigt, welche Mithilfe von TraV und Rfam identifiziert werden konnte. Die erkannte regulatorische RNA-Struktur ist in der Grafik mit einem grünen Pfeil markiert.

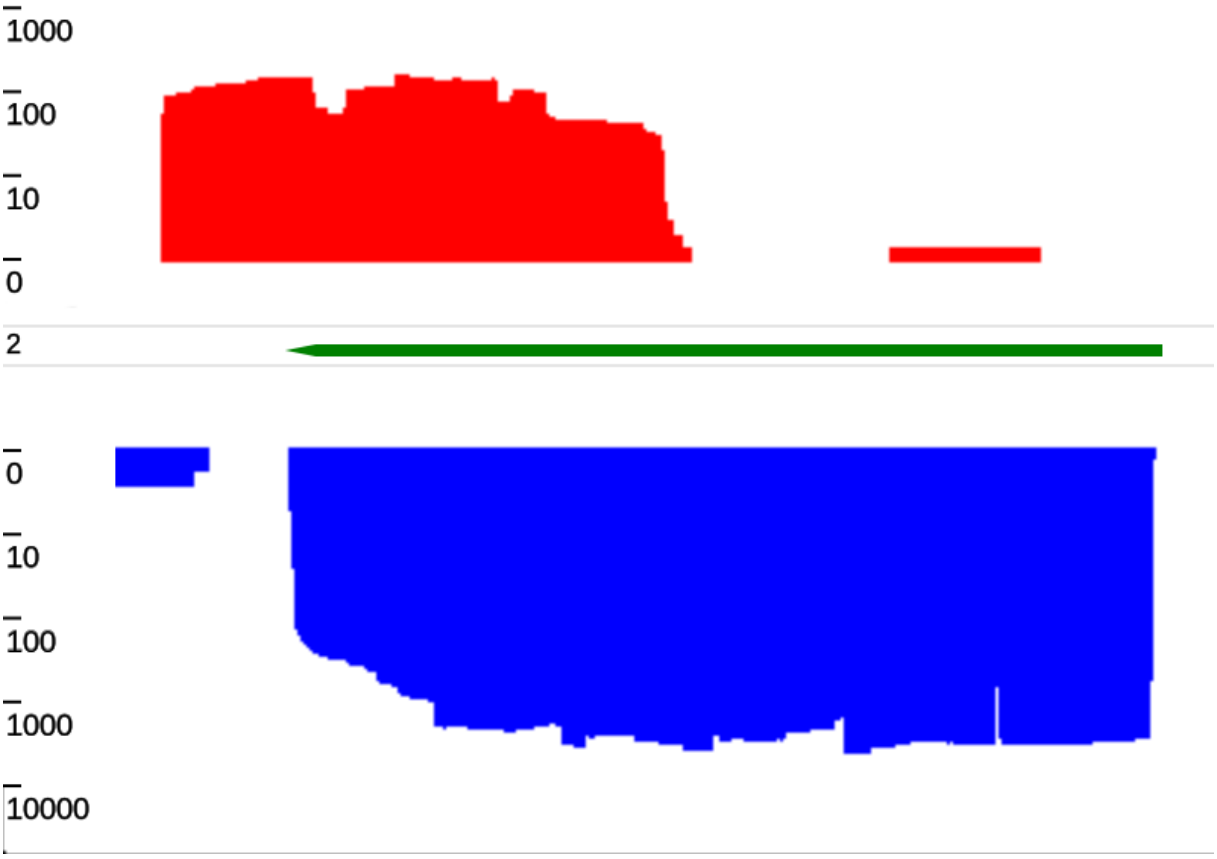

Abb. 18: Beispiele für *free transcripts*
Der rote wie auch der blaue Aktivitätsbereich zeigen transkriptionelle Aktivitäten ohne zugeordnete Annotationen. Der grüne Pfeil markiert einen Kandidaten für eine regulatorische RNA, *bsrG*/SR4

Solche *free transcripts* können regulatorische RNAs oder auch mRNAs mit fehlender Annotation sein. Daher sind weiterführende Analysen notwendig um die *features* die hinter den transkriptionellen Aktivitäten liegen zu identifizieren.

5.6.5 Suche nach *antisense* Transkripten

Antisense Aktivitäten sind transkriptionelle Aktivitäten, welche auf dem Gegenstrang zu einem *feature* liegen. Über die Komplementarität der Sequenzen von Strang und Gegenstrang können regulatorische Effekte entstehen (siehe 2.1.1). Aus diesem Grund wurde in TraV eine Suchmethode entwickelt, mit der solche *antisense* Transkripte identifiziert werden können. Der Algorithmus sucht dabei nach Bereichen transkriptioneller Aktivität, welche mit einem *feature* auf dem Gegenstrang teilweise überschneiden oder komplett überlappen. Abb. 19 zeigt ein solches Transkript, welches zu den Genen BLi00947 (*hypothetical protein*) komplett und BLi00948 (*putative antimicrobial peptide ABC exporter*) teilweise *antisense* liegt. Es wäre somit ein Kandidat für eine regulatorische *antisense* RNA.

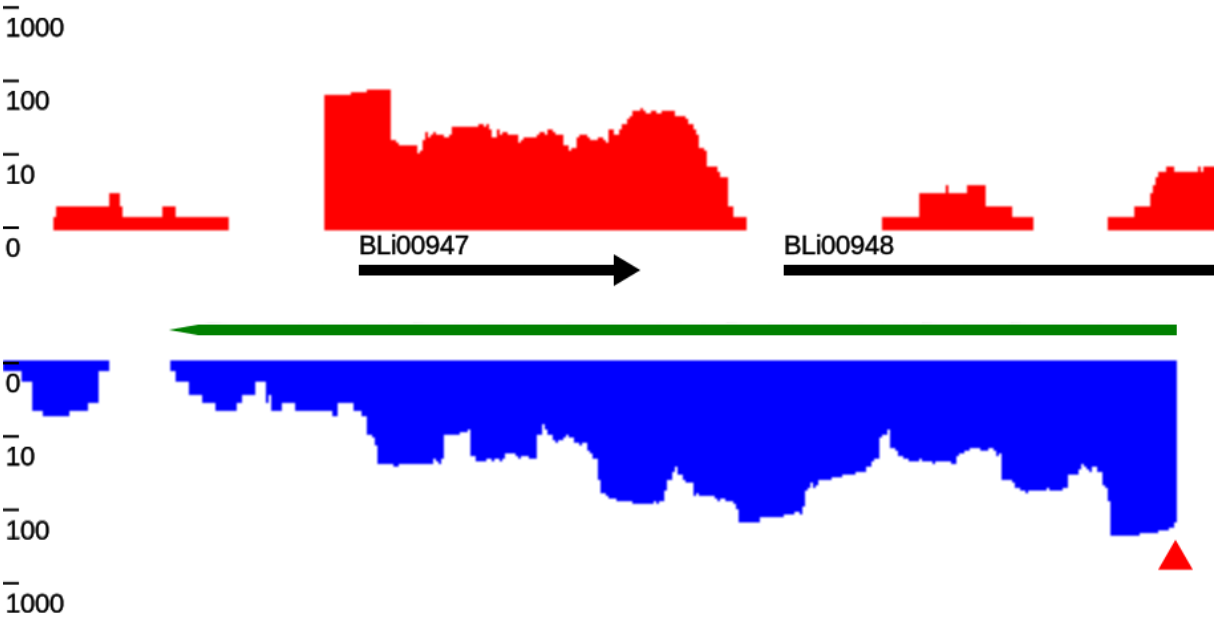

Abb. 19: Beispiel für ein *Antisense* RNA Transkript
Die blaue transkriptionelle Aktivität deutet auf ein RNA Transkript ohne genomische Annotation hin. Dieses Transkript liegt *antisense* zu den Genen BLi00947 und BLi00948. Der grüne Pfeil markiert das *antisense transcript*, der rote Pfeil markiert die TSS für dieses *antisense transcript*

Als Ergebnis der Suchmethode liefert TraV, neben den für die analytischen Methoden typischen Ergebnissen, zu den Kandidaten die prozentuale Abdeckung der einzelnen *antisense* Gene. Zusätzlich wird überprüft, ob das Transkript Teil eines annotierten Gens ist oder ob das Transkript ein *free transcript* ist. Das *antisense* Transkript im Beispiel von Abb. 19 ist z.B. ein *free transcript*.

5.7 Implementation der Analysemethoden

Die analytischen Methoden sind wie auch das Datenbankmanagement innerhalb der OmicsDatabase.jar Bibliothek implementiert. Für die Analysemethoden wurde innerhalb dieser Bibliothek ein Paket namens Analysis eingerichtet, das alle für die Analysen notwendigen Klassen enthält.

Alle Analysemethoden verwendet für die Erstellung ihrer Ergebnisse eine Klasse genannt featureBlock. Diese Klasse dient der Beschreibung der Ergebnisse und kann in ein GFF umgewandelt werden. Da alle Analysemethoden stets mehrere Kandidaten vorhersagen, werden als Ergebnis auch stets Listen von featureBlocks generiert. Alle Methoden (mit Ausnahme der *antisense* Transkriptsuche), können strangspezifisch oder strangunspezifisch verwendet werden.

Alle Methoden verwenden die gleichen Eingaben für ihre Analysen, nämlich ein Objekt vom Typ Replikon und einen Transkriptomdatensatz vom Typ TranscriptomeDataSet, welche beide Teil der OmicsDatabase.jar sind. Die Replikon.java Klasse liefert hierbei die

Annotationen und die Sequenzinformationen, die TranskriptomDataSet.java Klasse die Basenaktivitäten für die jeweiligen Analysen. Die Replikon.java Klasse bietet hierbei auch die Möglichkeit, die in ihr vorhandenen Annotationen mittels einer Liste von featureBlocks zu ergänzen oder zu ersetzen.

Die Suchmethoden sind in den folgenden Klassen implementiert. Alle Klassen bis auf die Suche nach *transcriptional start sites* und die Suche nach *antisense* Transkripten bieten ihre Suchmethoden als strangspezifische und strangunspezifische Varianten an.

- CalcNPKMs.java beinhaltet die Analysemethoden für die Berechnung von NPKM-Werten für Annotationen.
- FivePrimeExtFind.java und ThreePrimeExtFind.java beinhalten die Analysemethoden für die Suche nach 5'UTRs und 3'UTRs für Annotationen.
- FindFreeTranscripts.java beinhaltet die Methoden für die Suche nach *free transcripts.*
- FindAntisenseTranscripts.java implementiert die Suchmethode für *antisense* Transkripte.
- TSSpred.java implementiert die TSS-Suchmethode.

6 Auswertung der TraV Vorhersagen von *B. licheniformis* DSM13 RNA-Seq Daten aus industrieller Fermentation

Während der Entwicklung von TraV wurden die Transkriptomdaten von *Bacillus* licheniformis DSM13 in Kooperation mit Sandra Wiegand ausgewertet.

Das experimentelle Setup ist in Abb. 20 nach Wiegand *et al.* (Wiegand *et al.*, 2013) dargestellt. Insgesamt wurden fünf Proben zu verschiedenen Zeitpunkten der Fermentation genommen (römische Zahlen in Abb. 20). Die Kontrolle der Probenzeitpunkte geschah abhängig von Prozessparametern wie dem Sauerstoffpartialdruck, Acetat- und CO_2-Gehalt, da eine Bestimmung der Phase durch die Zelldichte aufgrund des industriellen Mediums nicht möglich ist. Die Fermentation wurde dreimalig repliziert. Diese Replikate werden als L, R und M bezeichnet.

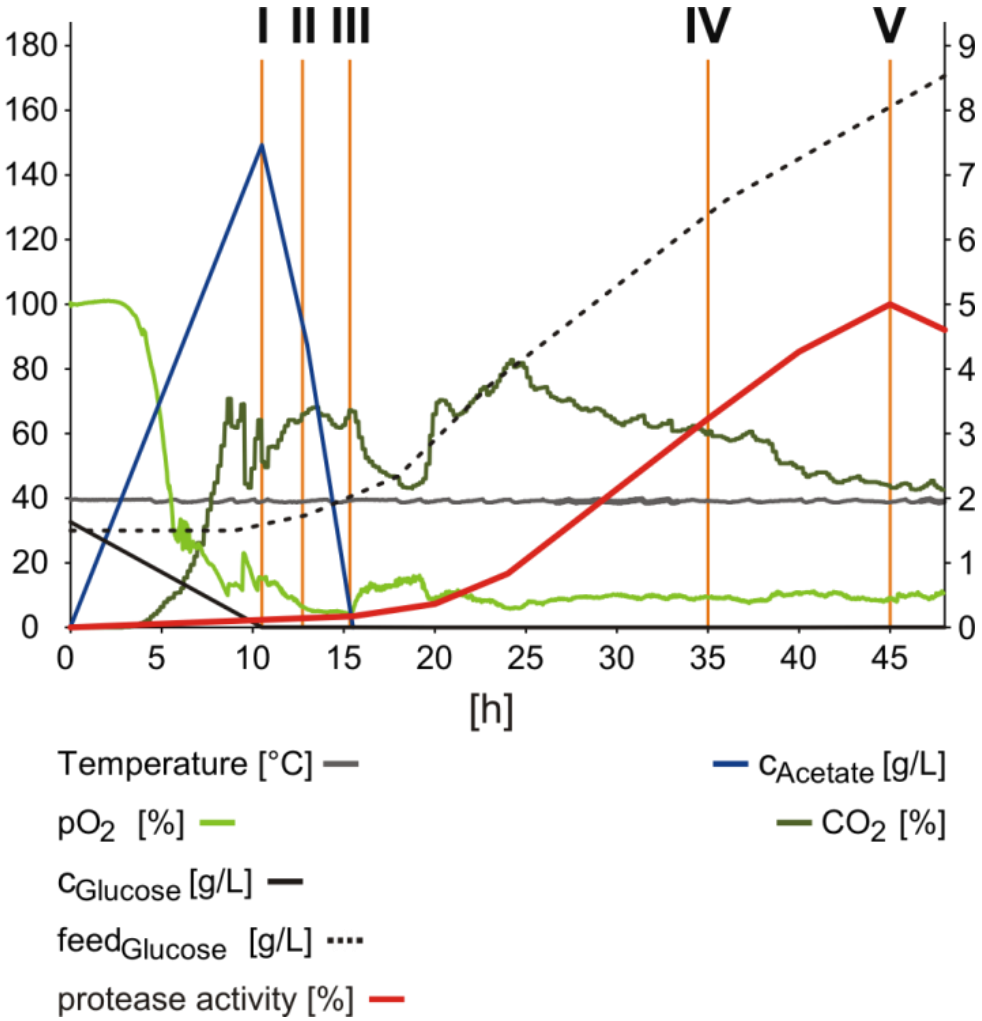

Abb. 20: Verlauf der industriellen Fermentation von *B. licheniformis* DSM13 nach Wiegand *et al.*
Die römischen Ziffern markieren die Probenzeitpunkte innerhalb der Fermentation. Ablauf der Fermentation wurde durch die Prozessparameter Acetatgehalt, Sauerstoffpartialdruck und CO_2-Gehalt gemessen

In Rahmen der Auswertung wurden die in 5.6 beschriebenen, analytischen Methoden designed, implementiert und evaluiert. Dieser Prozess bestand aus einer wechselseitigen Generierung und Auswertung der von den Analysemethoden erhaltenen Kandidatenlisten. Diese Ergebnisse wurden manuell kuriert um eventuell Grenzfälle, die von den Algorithmen nicht gefunden werden konnten, zu identifizieren und wenn möglich, zu berichtigen. Die 5‘ UTR, 3’UTR und *free transcripts* Kandidaten wurden zusätzlich mittels Interpro scan (Quevillon *et al.*, 2005) auf bekannte Peptidsignaturen überprüft, um so evtl. fehlende Annotationen zu identifizieren und nachzutragen.

Mit der TraV Methode wurden für die 5‘ und 3‘ UTRs 1404 und 1396 Kandidaten gefunden, wobei nach manueller Korrektur 1433 und 1365 Kandidaten verblieben (Wiegand *et al.*, 2013). Manuelle Korrekturen beinhalteten z.B. die Neuzuordnung eines *free transcripts* zu einer UTR wenn sich im Expressionsverhalten des Transkripts zeigt, dass es sehr nahe vor einem Gen endete oder *features* beinhaltet, die UTR-spezifisch sind. Entsprechende Transkripte wurden aus der Kandidatenliste der *free transcripts* entfernt. Für die *free transcripts* wurden 476 Kandidaten vorhergesagt, von denen 461 nach manueller Kuration übrig blieben. Für die *antisense transcripts* wurden 3777 Kandidaten vorhergesagt. Die Auswertung der *antisense transcript* Kandidaten war schwierig, da innerhalb des Genoms eine überraschend hohe Anzahl an *antisense* Transkripten gefunden wurde. Viele dieser *antisense* Transkripte sind kurz und verfügen über eine nur geringe Abdeckung. Außerdem scheint eine große Anzahl an 5‘ und 3’UTRs *antisense* zu Genen zu liegen. Im Fall der 3’UTRs scheint dies Aufgrund von unvollständiger Terminierung der Fall zu sein.

Innerhalb der Publikation wurden 855 der 3777 Kandidaten als potentielle *antisense transcripts* nach manueller Evaluation behalten. Das Ausschlussverfahren basiert auf einer Kombination der *antisense transcript* Vorhersagen mit den 3‘,5’UTR und *free transcripts* Vorhersagen sowie dem Ausschluss von *antisense transcripts* mit geringer Aktivität.

Die erhaltenen UTR und *free transcript* Kandidaten wurden mittels Infernal und der Kovarianzmodelle aus Rfam auf bekannte regulatorische RNA Strukturen überprüft. Anhand dieser Rfam Vergleiche konnten verschiedene *riboswitches* sowie ncRNAs identifiziert werden. Mittels TraV wurden am Beispiel der TPP, SAM und FMN *riboswitches* sowie der Vorhersagen von *bsrG* Toxin/Anti-toxin Systemen der genomische Kontext und das Expressionsverhalten dieser Kandidaten genauer betrachtet. Anhand der Rfam-Kovarianzmodelle wurden die gefundenen Strukturen der *riboswitches* miteinander verglichen. Dabei wurden die in Infernal erhaltenen Faltungsvorhersagen in VARNA (Darty *et al.*, 2009) visualisiert. Bei dieser Visualisierungsmethode wird das Maß der Übereinstimmung zwischen erwarteter Struktur durch das Kovarianzmodell und der Sequenz gezeigt. Daher sehen alle gezeigten Strukturen eines Modells nahezu gleich aus, da die Struktur vom

Modell bestimmt wird. Beim Abgleich der Sequenz mit der Struktur kann dann bestimmt werden, wie gut die Sequenz die vom Kovarianzmodell vorgegebene Faltungsstruktur einnimmt.

Neben den sRNAs konnten 3064 Kandidaten für Transkriptionelle Startpunkte (TSS) gefunden werden. Von diesen Kandidaten blieben nach manueller Kuration 1500 Kandidaten übrig. Diese Kuration basierte auf der Bestätigung der TSS Kandidaten durch die Replikate, Abgleich mit den 5'UTR und *free transcript* Listen sowie die TEX behandelten Datensätze (siehe Kapitel 4). Anhand dieser TSS Kandidaten wurde eine RNA-Seq geleitete Promotorvorhersage durchgeführt (siehe Kapitel 7).

Um mögliche Terminatoren vorherzusagen, wurde das Genom von *Bacillus licheniformis* DSM13 mittels TransTermHP (Kingsford *et al.*, 2007) durchsucht. Diese Vorhersagen werden in den Abbildungen in den folgenden Kapiteln integriert.

In den folgenden Unterkapiteln werden Beispiele für verschiedene, vorhergesagte *features* im Detail behandelt. Alle in diesen Kapiteln gezeigten Daten beziehen sich auf die M1 bis M5 Datensätze. Die Replikate L und R bestätigen idR. die gefundenen Ergebnisse. Tabellen mit den entsprechenden Ergebnissen für alle Replikate sind im Verzeichnis NPKMListen im Oberverzeichnis TraV_Vorhersagen der Daten-CD zu finden. Sollten die Replikate sich unterscheiden, werden diese Unterschiede entsprechend erwähnt.

Die den Auswertungen zugrunde liegenden Rohdaten befinden sich auf der Daten-CD im Oberverzeichnis Listen wobei das TraV_Vorhersagen Unterverzeichnis die vorhergesagten Kandidaten und das Rfam Verzeichnis die Rfam Vorhersagen und deren ausgewertete Strukturdaten beinhaltet.

6.1 *Thiamine-Pyrophosphate riboswitches* (TPP-*riboswitches*)

Thiamine-pyrophosphate ist ein Kofaktor des Kohlenstoff Metabolismus, wo es als Kofaktor der *pyruvate-dehydrogenase*, *pyruvate-decarboxylase* und α-*ketoglutarate-dehydrogenase* dient (Miranda-Ríos *et al.*, 2001). Insgesamt konnten vier TPP-*riboswitches* identifiziert werden.

Alle gefundenen *riboswitches* liegen *upstream* von Genen, welche in *thiamine* Biosynthese und Transport beteiligt sind.

- Das *thiC* Gen kodiert für die *phosphomethylpyrimidinesynthase*, welche an der Bildung von *4-Amino-5-hydroxymethyl-2-methylpyrimidine* (HMP-P), einer Vorstufe von *thiamindiphosphaet*, beteiligt ist.
- Das Gen *thiT* kodiert für einen aktiven *thiamine* Transporter.

- Das Operon *tenAI/thiOSGFD* (Toms *et al.*, 2005) kodiert Gene für die *thiazole* Biosynthese, einer Vorstufe von *thiamine* sowie ein Gen für die Bildung von *4-amino-2-methyl-5-phosphomethylpyrimidine* (HMP-PP).
- Die Gene *tenAI* kodieren eine *thiaminase* und eine *thiazole-tautomerase* (Hazra *et al.*, 2011). Die Gene *thiOSGFD* kodieren für eine *glycine-oxidase* (*thiO*), ein *sulfur carrier protein (thiS),* eine *thiazole synthase(thiG)*, eine *hydroxymethylpyrimidine/ phosphomethylpyrimidine kinase* (*thiD*). Die Funktion von *thiF* ist derzeit unbekannt.
- Das *thiVWX* Operon kodiert für einen *thiamine* ABC Transporter.

Tabelle 13 listet die gefundenen *riboswitches* mitsamt ihrer *scores* für das Kovarianzmodell auf. Der *trusted-cutoff* für das Kovarianzmodell beträgt 30,1. Diesen *score* überschreiten vier der sechs Treffer. Für alle Treffer, die oberhalb des *trusted-cutoff* lagen gibt es korrespondierende Kandidaten aus den 5'UTR Vorhersagen. Die zwei Treffer unterhalb des *trusted-cutoff* entstammen dagegen den *free transcript* Vorhersagen.

Tabelle 13: Treffer des Rfam Kovarianzmodells für TPP *riboswitches*

Phase	**Name des Kandidaten**	**Koordinaten im Genom**	**Score**
1	UTR5_301	941.981+942.290	81,76
	UTR5_1054	3.125.034+3.125.259	62,87
	UTR5_1369	4.036.819-4.037.250	85,84
2	UTR5_289	941.947+942.290	81,76
	UTR5_1028	3.125.034+3.125.259	62,87
3	UTR5_285	941.980+942.290	81,76
	UTR5_994	3.125.034+3.125.259	62,87
	UTR5_1290	4.036.819-4.037.069	85,84
4	UTR5_315	942.026+942.290	81,76
	UTR5_461	1.271.163+1.271.391	82,90
	UTR5_1182	3.125.032+3.125.259	62,87
	UTR5_1533	4.036.819-4.037.084	85,84
	sRNA_387	2.741.361+2.741.721	8,47
	sRNA_549	4.036.963+4.037.043	25,87
5	UTR5_310	941.981+942.290	81,76
	UTR5_448	1.271.165+1.271.391	82,90

	UTR5_1130	3.125.034+3.125.259	62,87
	UTR5_1456	4.036.819-4.037.127	85,84
	sRNA_317	2.741.361+2.741.628	8,47

Abb. 21 (verändert nach Sonenshein *et al.*, 2002) gibt eine grafische Übersicht über die TPP Biosynthese sowie der *riboswitch* kontrollierten Gene. In der Abb. 22 sind die transkriptionellen Aktivitäten der identifizierten TPP-*riboswitches* im genetischen Kontext abgebildet.

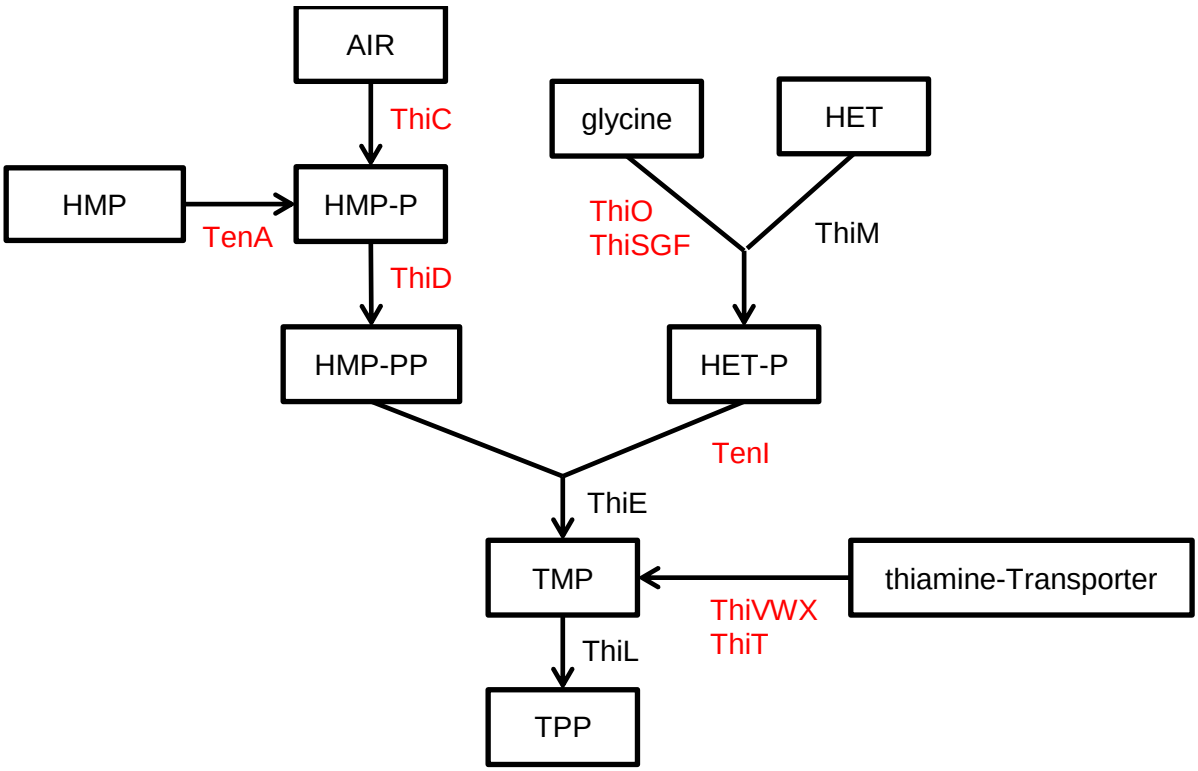

Abb. 21: Schematische Übersicht über die TPP *denovo* Synthese

Die mRNAs der rot markierten Proteine besitzen TPP *riboswitches.*
AIR: *Aminoimidizole ribotide*
HMP: *Hydroxymethylpyrimidine*
HMP-P: *Hydroxymethylpyrimidine phosphate*
HMP-PP: *Hydroxymethylpyrimidine pyrophosphate*
HET: *Hydroxyethylthiazole*
HET-P: *Hydroxyethylthiazole phosphate*
TMP: *Thiamine monophosphate*
TPP: *Thiamine pyrophospate*

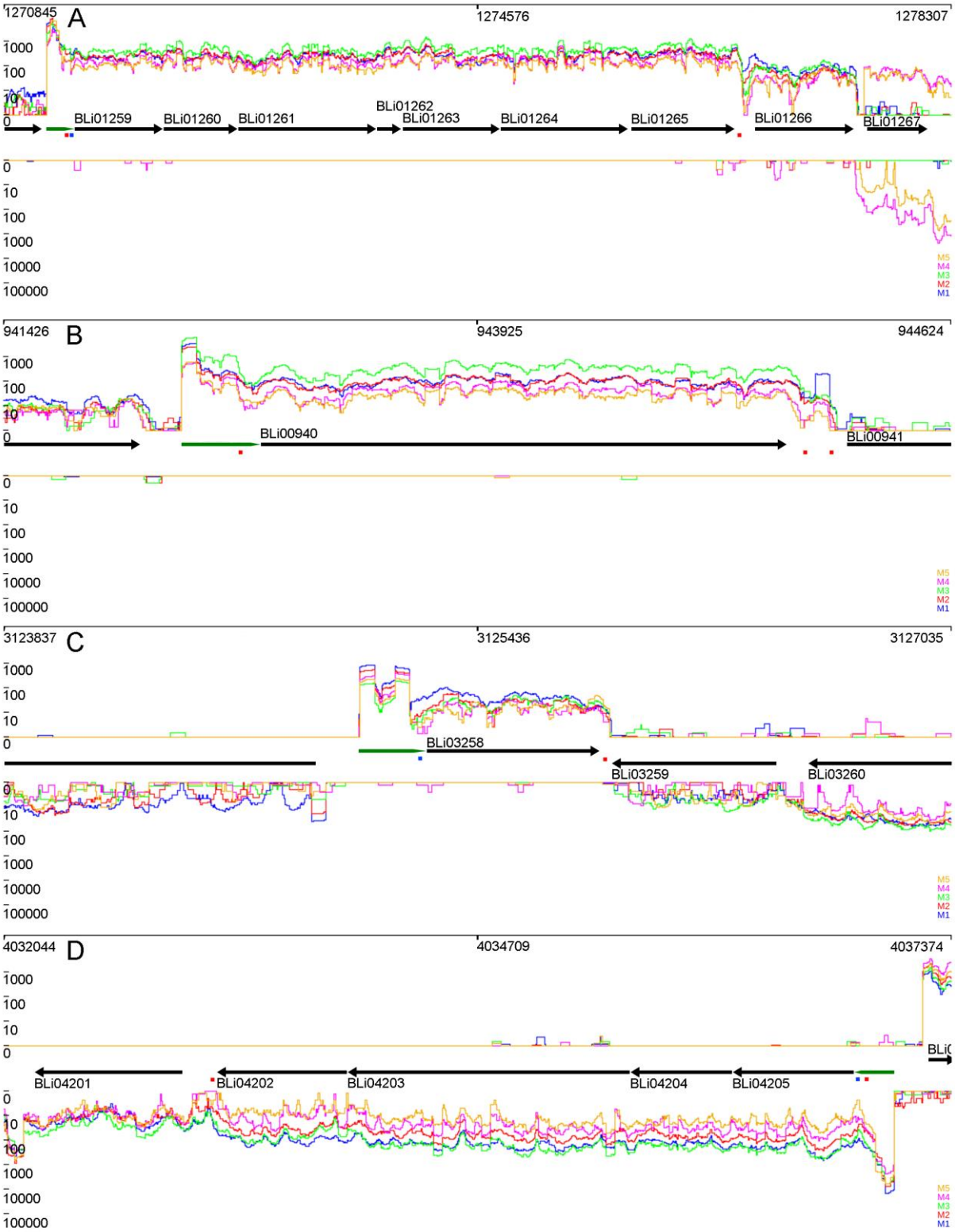

Abb. 22: Transkriptionale Aktivitäten von vorhergesagten TPP-*riboswitches* in den Phasen M1-M5 im genomischen Kontext
Mit Rfam vorvorhergesagte TPP-*riboswitches* sind mit grünen Pfeilen markiert. Grafik A zeigt den Kontext des *tenAI/thiOSGFD* Operons. Grafik B zeigt den Kontext des *thiC* Gens. Grafik C zeigt den Kontext des *thiT* Gens. Grafik D zeigt den Kontext des *thiVWX* Operons. Mit TransTermHP vorhergesagte Terminatorstrukturen sind mit roten Kästchen markiert. Manuell identifizierte Shine-Dalgarno Sequenzen sind mit blauen Kästchen markiert

Anhand der Expressionsprofile lässt sich sagen, dass alle TPP-*riboswitch* kontrollierten Gene Transkripte vorweisen. Somit unterdrückt keiner der hier betrachteten

TPP-*riboswitches* die Expression von Genen vollständig. Die Menge an Transkripten und damit die Expressionsstärke der *riboswitches* variiert je nach Gen und Wachstumsphase stark, wobei wahrscheinlich Effekte wie Ausbildung der Struktur und damit Stabilisierung der RNA eine Rolle spielen (Belasco and Higgins, 1988; Storz *et al.*, 2004). Interessant ist, dass mittels TransTermHP in den Endbereichen von dreien der vier *riboswitches* potentielle Terminatoren vorhergesagt werden konnten. *riboswitches* bilden solche Terminatorstrukturen aus um die Transkription zu unterbrechen und so ihren regulatorischen Effekt zu bewirken. Tabelle 14 gibt einen Überblick über die NPKMs der TPP *riboswitches*.

Tabelle 14: NPKM Werte für die vorhergesagten TPP *riboswitches* in den Phasen M1-M5

riboswitch	**M1 NPKM**	**M2 NPKM**	**M3 NPKM**	**M4 NPKM**	**M5 NPKM**
thiT	2771	1794	603	1211	826
thiVWX	35303	18556	15657	6912	18038
thiC	4630	3481	9434	883	920
tenAI/thiOSGFD	13355	13567	10018	5025	11541

Die Tabelle 15 zeigt die NPKM-Werte der jeweiligen Operons, Tabelle 16 gibt die Verhältnisse zwischen der Expressionsstärke von *riboswitch* und jeweiligem Operon wieder.

Tabelle 15: NPKM Werte der TPP *riboswitch* regulierten Operons in den Phasen M1-M5

Operon	**M1 NPKM**	**M2 NPKM**	**M3 NPKM**	**M4 NPKM**	**M5 NPKM**
thiT	212	112	109	67	73
thiVWX	979	426	1152	195	98
thiC	604	559	1843	252	162
tenAI/thiOSGFD	1608	1686	2915	1049	791

Tabelle 16: Verhältnisse der TPP-*riboswitch*-Expressionsstärken zu den jeweiligen Operons

Operon	**M1 Ratio**	**M2 Ratio**	**M3 Ratio**	**M4 Ratio**	**M5 Ratio**
thiT	13,07	16,02	5,53	18,07	11,32
thiVWX	36,06	43,56	13,59	35,45	184,06
thiC	7,67	6,23	5,12	3,50	5,68
tenAI/thiOSGFD	8,31	8,05	3,44	4,79	14,59

Vergleicht man diese TPP *riboswitch* Expressionsstärken mit den jeweiligen Operons, die sie kontrollieren, zeigt sich, dass die *riboswitches* einen Einfluss auf die Expressionsstärke haben. Ein kleines Verhältnis zwischen transkriptionaler Aktivität von*riboswitch* und Operon deutet darauf hin, dass es zwischen *riboswitch* und Operon geringe Unterschiede in der Expressionsstärke gibt womit der *riboswitch* die Bildung der mRNA weniger reprimiert. Ist das Verhältnis groß, deutet dies darauf hin, dass der *riboswitch* die Expression der folgende

Gene stark blockiert. Das *thiT* Gen weist in der M3 Phase den kleinsten Ratio auf, wogegen er während der Phasen M1, M2 sowie M4 und M5 höher liegt. Das *thiVWX* Operon verhält sich vergleichbar zum *thiT* Gen, wobei es während der M5 Phase einen höheren Ratio aufweist. Das Operon *tenAI/thiOSGFD* zeigt ebenfalls während der M3 Phase den geringsten Ratio und verhält sich ähnlich wie *thiVWX* und *thiT.* Das *thiC* weißt von allen verglichenen Operons die geringste Schwankung in den Expressionsratios auf wobei der geringste Ratio während der M4 Phase erreicht wird und nicht wie bei den anderen Operons in Phase M3.

Das Produkt von *thiC* scheint vor allem während der exponentiellen Wachstumsphase wichtig zu sein. Im Gegensatz zu den *thiVWX* und *TenAI/thiOSGFD* Genen steigt sein Ratio in den stationären Phasen nicht an. *ThiVWX* wird in der späten stationären Phase stark reprimiert was ein Hinweis darauf sein könnte, dass entweder weniger *thiamine* in der Zelle benötigt wird oder das Medium nicht mehr genug *thiamine* für den Import anbietet. Interessanterweise behält der zweite Transporter *thiT* einen gewissen Grad an Aktivität. Das *tenAI/thiOSGFD* Operon enthält Gene für *thiamine* Wiedergewinnung sowie den Großteil der Gene für die *denovo* Synthese von *thiamine* und ist wie *thiC* anscheinend besonders während der exponentiellen Wachstumphase in Verwendung. *TenAI/thiOSGFD* zeigt aber eine große Aktivität und geringe Ratios in allen Phasen. Daraus lässt sich schließen dass *thiamine* besonders während des exponentiellen Wachstums benötigt wird und dass anscheinend während der späten stationären Phase weniger *thiamine* importiert wird.

In Abb. 23 sind die Strukturvorhersagen der TPP *riboswitches* durch VARNA visualisiert. Stärkere Abweichungen vom Kovarianzmodell gibt es nur beim *thiT riboswitch*, welchem die Basen für den *loop* ab Base 22 bis 27 fehlen. Auch die *thiVWX* und *thiC riboswitches* weisen in diesem *loop* Lücken auf, jedoch nicht im Ausmaß des *thiT riboswitches*.

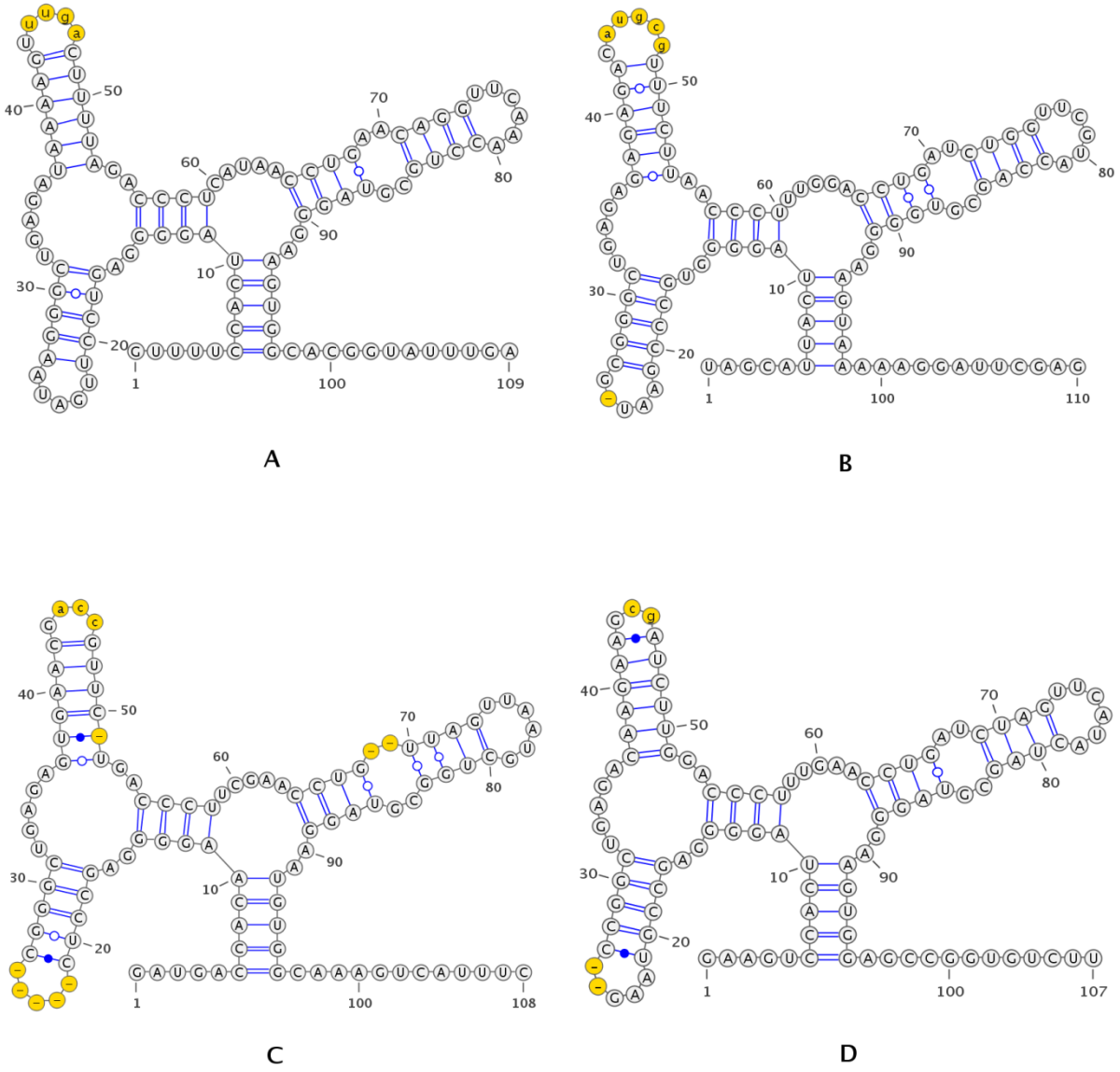

Abb. 23: Vergleich der vorhergesagten Riboswitchstrukturen mit dem Rfam Kovarianzmodell mittels VARNA

Grafik A zeigt den *tenAI/thiOSGFD riboswitch*. Grafik B zeigt den *thiC riboswitch*. Grafik C zeigt den *thiT riboswitch*. Grafik D zeigt den *thiVWX riboswitch*. Hierbei ist zu beachten, dass stets die Struktur des Kovarianzmodells gezeigt wird. Das Maß der Übereinstimmung mit der Struktur wird durch den Vergleich der Sequenz mit der erwarteten Struktur bestimmt. Stabile Basenpaarungen werden durch Striche zwischen den Paaren dargestellt. Unpassende oder instabile Paarungen werden durch Kreise dargestellt. Unterschiede zum Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell werden durch kleingeschriebene Basen angegeben

6.2 S-Adenosylmethionine *riboswitche* (SAM-*riboswitche*)

S-Adenosylmethionine (SAM) ist ein Kofaktor und dient als Methylgruppendonor in der Biosynthese von *polyamine* und *biotine* (Sonenshein *et al.*, 2002). Eine Reihe von SAM abhängigen Operons wird über SAM *riboswitches* reguliert (Winkler and Breaker, 2005). In den *B. licheniformis* DSM13 Transkriptomdaten konnten zehn SAM-*riboswitches* vorhergesagt werden. Diese *riboswitches* liegen vor Genen, die an der Biosynthese von *methionine*, *cystheine* und *S-Adenosylmethionine* beteiligt sind, sowie vor Transportern für *methionine*.

- Das Gen *metK* kodiert für eine *S-adenosylmethionine synthase* (Grundy and Henkin, 1998)
- Das Operon *metIC* kodiert für eine *cystathione* γ*-synthase* und eine *cystathione* β*-lyase* (Auger *et al.*, 2002)
- Das Gen *yxjG* kodiert für eine *methionine synthase* (Grundy and Henkin, 1998)
- Das Operon *metQ1N1P1* sowie *metQ2N2P2* kodieren für jeweils einen *methionine transporter*.
- Das Gen BLi03178 kodiert für eine putative *methionine transporter* Komponente.
- Die Genprodukte des *mtnWB(X)D* Operons sind an der Methionin-Wiedergewinnung aus Methylthioadenosin beteiligt (Grundy and Henkin, 1998). Das *mtnW* Gen kodiert für eine *2,3-diketo-5-methylthiopentyl-1-phosphate enolase*. Das *mtnB/mtnX* Gen ist eine Fusion aus zwei Genen welche eine *ribulose-5-phosphate epimerase* und eine *2-hydroxy-3-keto-5-methylthiopentenyl-1-phosphate phosphatase* kodieren. Das *mtnD* Gen kodiert eine *1,2,-dihydroxy-3-keto-5-methylthiopentene dioxygenase*(Sekowska and Danchin, 2002) (Michna *et al.*, 2014).
- Das Operon *mtnKA* kodiert für eine *5-methylthioribose kinase* und eine *5-methylthioribose-1-phosphate isomerase*. Beide Enzyme sind Teil der Wiedergewinnung von *methionine* aus Nebenprodukten der Polyamin Synthese (Nakano *et al.*, 2014).
- Das Operon BLi01777-*cysH1P1/sat/cysC* kodiert Gene der Cysteinsynthese (Mansilla *et al.*, 2000). Die Gen *cysH1P1* kodieren für eine *phosphoadenosine phosphosulfate sulfotransferase* und eine *sulfate permease.* Das Gen *sat* kodiert für eine *sulfate adenylyltransferase*. Das Gen *cysC* kodiert für eine *adenylyl-sulfate kinase* (Mansilla *et al.*, 2000).

  Mansilla et *al.* beschreiben das *cysH* Operon in *B. subtilis* als ein 7 Gene umfassendes Operon welches zwei alternative Transkripte ausbildet, ein 6,1 kb großes Transkript das alle Gene beinhaltet und ein 4,8 kb Transkript, welches die Gene *cysHP/sat/cysCG* beinhalten würde. Anhand unserer RNA-seq Daten lässt sich

zeigen, dass *B. licheniformis* DSM13 dieses Schema anscheinend nicht einhält und stattdessen zwei Operons zu bilden scheint (BLi01777-*cysH1P1/sat/cysC* und *cysG/sirBC*) welche beide über potentielle Promotoren verfügen. Außerdem ist für *B. subtilis* kein hypothetisches Gen vor *cysH* annotiert. Die potentiellen Promotoren sind in Abb. 28CD abgebildet.

Das Operon *yitJ/metH* kodiert für eine *methionine synthase* (Grundy and Henkin, 1998).

Tabelle 17 gibt einen Überblick über die mittels TraV und Rfam vorhergesagten SAM-*riboswitches*. Diese Tabelle ist der Übersichtlichkeit wegen auf die Treffer mit einem *score* oberhalb des *trusted-cutoffs* beschränkt. Neben den hier aufgelisteten Treffern gab es verschiedene Treffer mit anderen SAM Kovarianzmodellen, welche spezifisch für andere Klassen von Prokaryoten sind. Eine volle Übersicht ist in SAM_parsedRes.xlsx auf der Daten-CD zu finden. Der *yitJ/MetH riboswitch* wurde durch manuelle Untersuchung von SAM assoziierten Genen gefunden, da das Expressionsverhalten des vorhergehenden Genes die analytischen Methoden von TraV blockiert.

Abb. 24 gibt eine Übersicht über die SAM, *methionine* und *cysteine* Biosynthese und markiert die Gene, welche unter der Kontrolle von *riboswitches* stehen. Grafik verändert nach Tomsic *et al*. (Tomsic *et al.*, 2008).

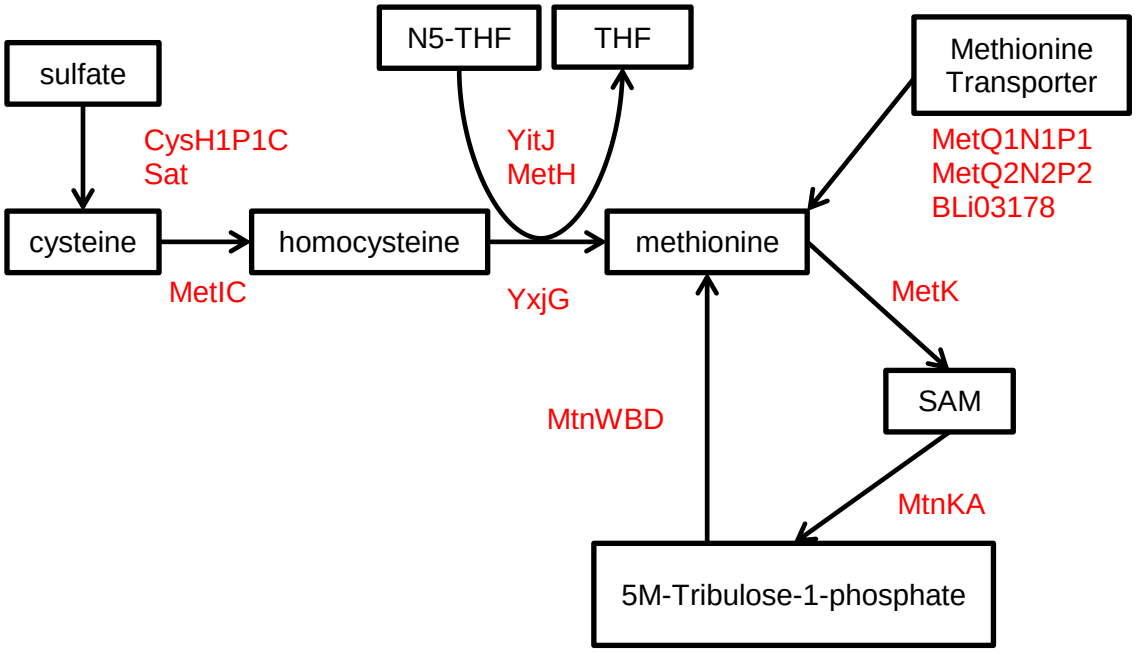

Abb. 24: Übersicht über die SAM Biosynthese
In Rot markierte Genprodukte besitzen einen in ihrer mRNA einen vorhergesagten SAM-*riboswitch*
THF: *Tetrahydrofolate*
SAM: *S-adenosylmethionine*

Tabelle 17: Vorhersagen für SAM-*riboswitches* mittels Rfam Kovarianzmodellen

Phase	Name	Koordinaten	Score	Phase	Name	Koordinaten	Score
1	>UTR5_313	969.706+969.939	72,00	2	>UTR5_420	1.292.056+1.292.289	92,83
	>UTR5_441	1.292.001+1.292.289	92,83		>UTR5_586	1.735.560+1.735.756	85,93
	>UTR5_598	1.735.479+1.735.756	85,93		>UTR5_1009	3.063.553-3.063.761	85,25
	>UTR5_1031	3.063.553-3.063.753	85,25		>UTR5_1015	3.080.284-3.080.765	71,63
	>UTR5_1037	3.080.284-3.080.765	71,63		>UTR5_1095	3.302.393-3.302.657	93,72
	>UTR5_1125	3.302.393-3.302.657	93,72		>UTR5_1335	4.014.331+4.014.539	94,08
	>UTR5_1366	4.014.331+4.014.539	94,08		>UTR3_489	1.487.319-1.487.589	76,93
	>UTR3_488	1.487.280-1.487.589	76,93		>sRNA_97	969.677+969.916	72,00

Phase	Name	Koordinaten	Score	Phase	Name	Koordinaten	Score
3	>UTR5_410	1.292.074+1.292.289	92,83	4	>UTR5_466	1.292.047+1.292.289	92,83
	>UTR5_568	1.735.560+1.735.756	85,93		>UTR5_632	1.735.549+1.735.756	85,93
	>UTR5_978	3.080.284-3.080.765	71,63		>UTR5_1163	3.080.284-3.080.765	71,63
	>UTR5_1060	3.302.393-3.302.688	93,72		>UTR5_1268	3.302.393-3.302.657	93,72
	>UTR5_1287	4.014.316+4.014.539	94,08		>UTR5_1529	4.014.330+4.014.539	94,08
	>UTR3_492	1.487.318-1.487.589	76,93		>UTR3_514	1.487.318-1.487.589	76,93
	>UTR3_493	1.489.656+1.489.837	84,06		>sRNA_142	969.706+969.918	72,00
	>sRNA_95	969.636+969.909	72,00		>sRNA_434	3.063.577-3.063.750	85,25

Phase	Name	Koordinaten	Score	Phase	Name	Koordinaten	Score
5	>UTR5_454	1.291.984+1.292.289	92,83	1-5	Manuelle Vorhersage	1.207.567-1.207.797	99,21
	>UTR5_618	1.735.561+1.735.756	85,93				
	>UTR5_1109	3.080.284-3.080.764	71,63				
	>UTR5_1202	3.302.393-3.302.657	93,72				
	>UTR5_1452	4.014.331+4.014.539	94,08				
	>UTR3_512	1.487.310-1.487.589	76,93				
	>sRNA_105	969.706+969.918	72,00				
	>sRNA_355	3.063.579-3.063.781	85,25				

Im Vergleich zu den TPP-*riboswitches* zeigen die SAM-*riboswitches* eine stärkere Regulation der Transkription. Eine genaue Benennung der aktivsten Phase ist schwierig, da die Replikate sich in ihren Aussagen in den Phasen 1 und 2 unterscheiden (siehe P15_SAMs.tsv unter NPKMListen im Anhang). Betrachtet man die NPKM-Werte der Replikate, scheinen die *riboswitches* in den Phasen 1 und 2 am stärksten exprimiert zu werden. Die NPKM-Werte für die *riboswitches* aus M1-M5 sind in Tabelle 18 aufgelistet. Die Gene dagegen zeigen relativ stabile Mengen an Transkripten über die Phasen M1 bis M5, erkennbar an den NPKMs in Tabelle 19 wobei auch hier Phase 1 (bzw. 2 in den anderen Replikaten) die Phase mit der größten Aktivität zu sein scheint.

Betrachtet man die Verhältnisse zwischen NPKMs von *riboswitch* und Operon in Tabelle 20, wird deutlich dass es einige Operons gibt, in denen die *riboswitches* stark regulieren (wie z.B. *metIC)*. Dies geht bis zu einer nahezu Unterdrückung der Expression, wie bei den *mtnKA*, *metQ1N1P1* und BLi3178 Operons zu erkennen ist. Die Operons *metK*, *cysH1P1/sat/cysC* und *mtnWBD* scheinen dagegen weniger stark reguliert zu sein, erkennbar an den kleineren Ratios in den Phasen M3-M5 bei *cysH1P1/sat/cysC* und *mtnWBD,* sowie Phasen M1-M5 bei *metK.* Trotz der starken Unterschiede in den NPKM-Werten kommen die Replikate bei den Verhältnissen zu ähnlichen Aussagen, mit der Eingangs erwähnten Schwankung zwischen Phase 1 und Phase 2.

Tabelle 18: NPKM-Werte der vorhergesagten SAM-*riboswitches* in Phasen M1-M5

riboswitch	**M1 NPKM**	**M2 NPKM**	**M3 NPKM**	**M4 NPKM**	**M5 NPKM**
metIC	22961	10986	2027	1698	2390
cysH1P1/sat/cysC	4036	1137	992	321	295
metK	362	215	151	64	89
metQ2N2P2	8073	4114	2439	1528	2035
yxjG	9291	3206	5063	2084	5421
mtnKA	2231	1564	1628	2415	1812
metQ1N1P1	480	237	127	133	231
BLi03178	1705	467	183	157	166
mtnWBD	3335	911	229	262	406
yitJ/metH	4347	2662	2543	2482	3258

Tabelle 19: NPKM-Werte der SAM-*riboswitch* regulierten Operons in Phasen M1-M5

Operon	M1 NPKM	M2 NPKM	M3 NPKM	M4 NPKM	M5 NPKM
metIC	32	15	17	29	25
cysH1P1/sat/cysC	211	55	171	123	71
metK	278	156	185	101	90
metQ2N2P2	261	34	71	55	41
yxjG	107	16	27	27	30
mtnKA	10	2	3	3	4
metQ1N1P1	8	2	3	2	3
BLi03178	12	3	1	3	2
mtnWBD	36	11	38	51	68
yitJ/metH	208	126	158	166	209

Tabelle 20: Verhältnisse der SAM-*riboswitch*-Expressionsstärken zu den jeweiligen Operons

riboswitch	M1 Ratio	M2 Ratio	M3 Ratio	M4 Ratio	M5 Ratio
metIC	717,53	732,40	119,24	58,55	95,60
cysH1P1/sat/cysC	19,13	20,67	5,80	2,61	4,15
metK	1,30	1,38	0,82	0,63	0,99
metQ2N2P2	30,93	121,00	34,35	27,78	49,63
yxjG	86,83	200,38	187,52	77,19	180,70
mtnKA	223,10	782,00	542,67	805,00	453,00
metQ1N1P1	60,00	118,50	42,33	66,50	77,00
BLi03178	142,08	155,67	183,00	52,33	83,00
mtnWBD	92,64	82,82	6,03	5,14	5,97
yitJ/metH	20,90	21,13	16,09	14,95	15,59

Abb. 25 zeigt die Kontexte sowie die Expressionsprofile der Gene welche an der Methionin- und SAM-Synthese beteiligt sind. Die erhöhte Aktivität in der Phase M1 ist bei den Genen *yxjG* sowie *metK* gut erkennbar. Das *yitJ/metH* Operon weist viele Unterbrechungen der Transkriptionsaktivität bei relativ hoher Abdeckung auf.

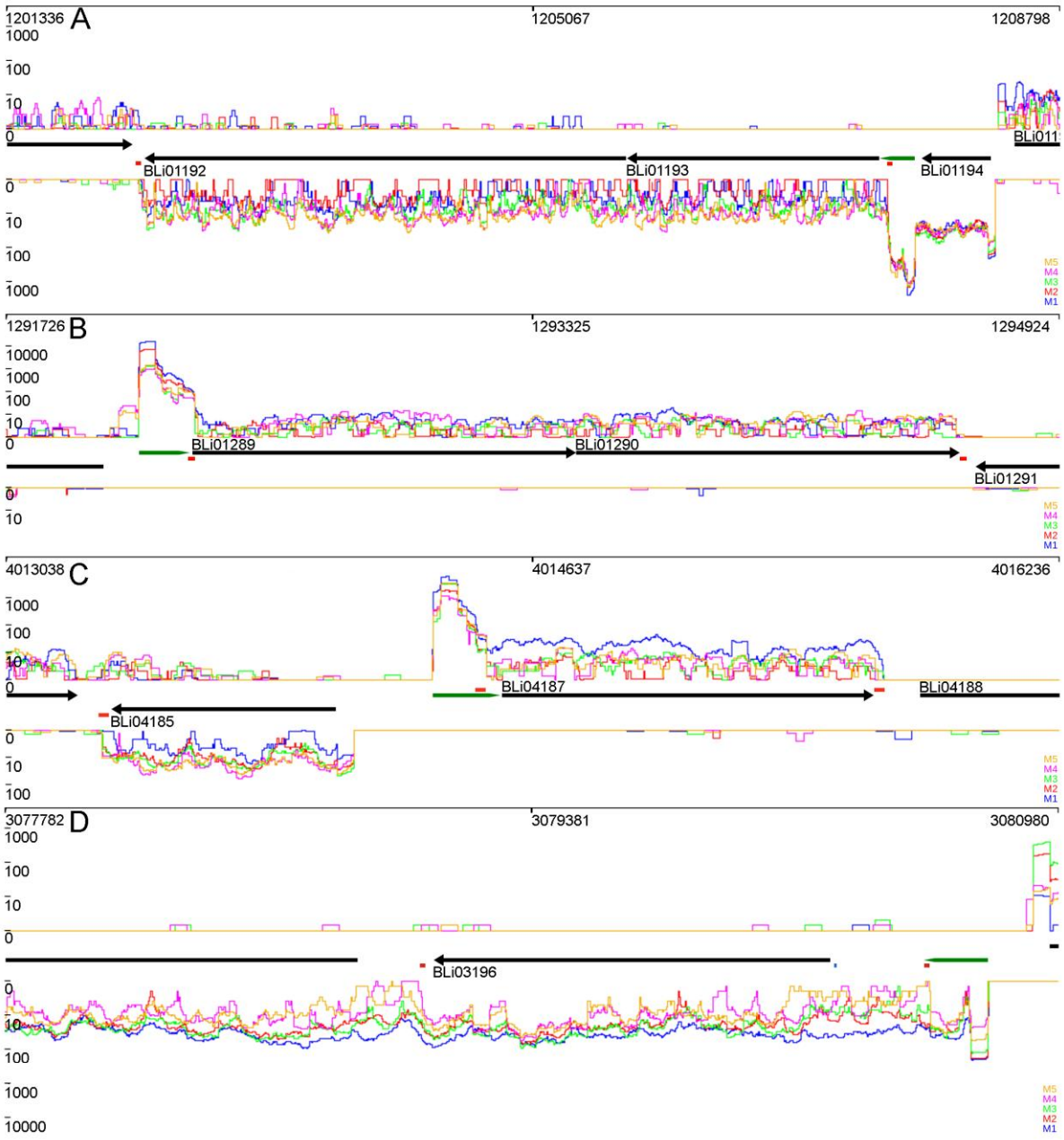

Abb. 25: Transkriptionale Aktivitäten von vorhergesagten SAM-*riboswitches* für Methionin- und SAM-Synthesegene im genomischen Kontext
Grafik A zeigt den Kontext des *yitJ/metH* Operons. Grafik B zeigt den Kontext des *metIC* Operons. Grafik C zeigt den Kontext des *yxjG* Gens. Grafik D zeigt den Kontext des *metK* Gnes. Mit Rfam vorhergesagte *riboswitches* sind mit grünen Pfeilen markiert. Rote Kästchen markieren mit TransTermHP vorhergesagte Terminatoren, blaue Kästchen markieren manuell vorhergesagte Shine-Dalgarno Sequenzen

Abb. 26 zeigt die transkriptionellen Aktivitäten der Methionintransportergene im genomischen Kontext. Auffällig ist, dass das *metQ1N1P1* Operon sowie das BLi03178 Gen inaktiv zu sein scheinen. Sie zeigen eine sehr geringe Aktivität ohne durchgängiges Transkript. Nur das *metQ2N2P2* Operon scheint aktiv zu sein wobei sich Phase M1 stark von den anderen Phasen abhebt.

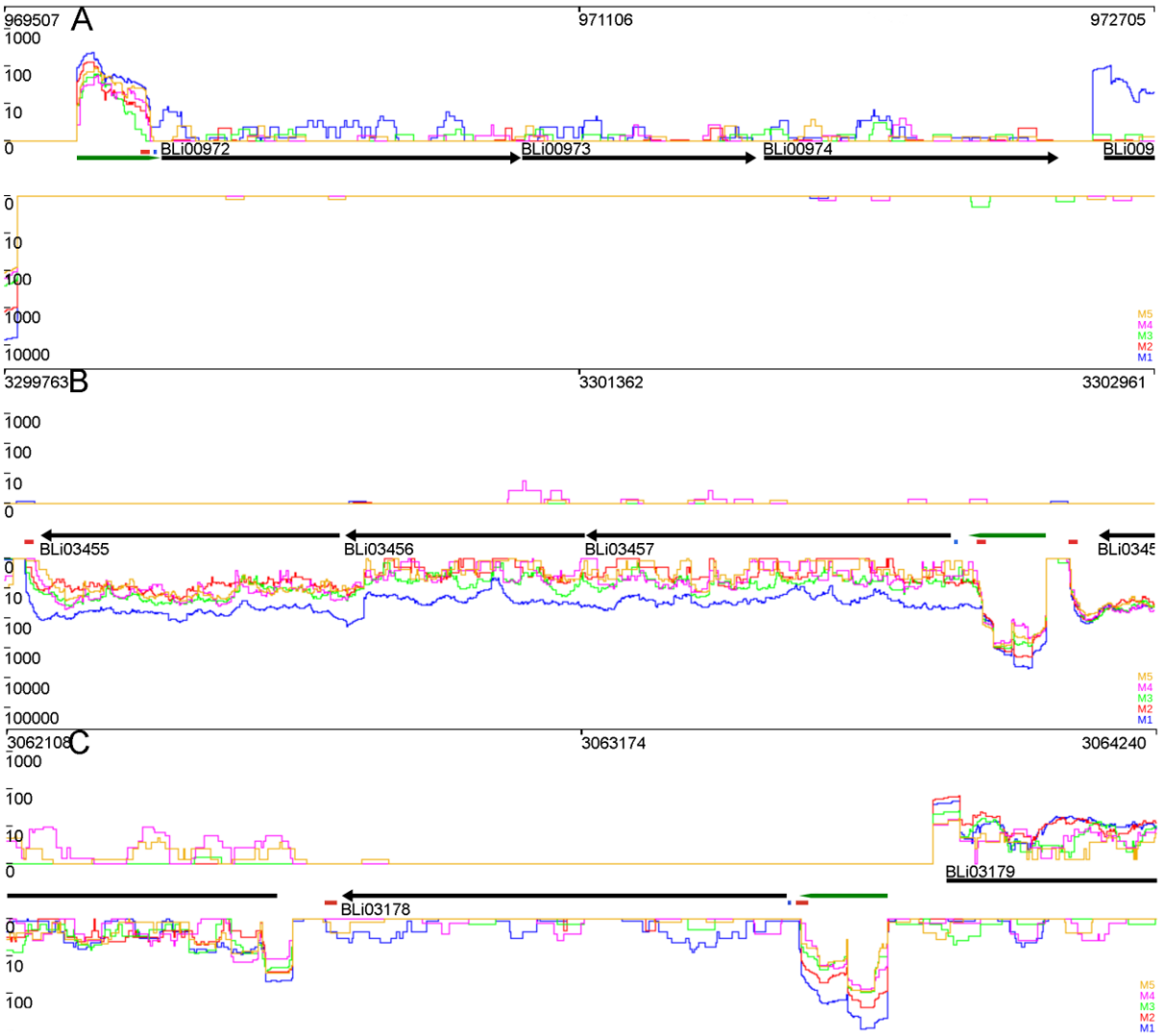

Abb. 26: Transkriptionale Aktivitäten von vorhergesagten SAM-*riboswitches* für Methionintransporter im genomischen Kontext
Grafik A zeigt den Kontext des *metQ1N1P1* Operons. Grafik B zeigt den Kontext des *metQ2N2P2* Operons. Grafik C zeigt den Kontext des BLi03178 Gens. Rote Kästchen markieren mit TransTermHP vorhergesagte Terminatoren, blaue Kästchen markieren manuell vorhergesagte Shine-Dalgarno Sequenzen

Abb. 27 zeigt die Kontexte der SAM-Metabolismusgene vor denen *riboswitches* vorhergesagt wurden. Beide dargestellten Operons zeigen sehr geringe Aktivität, wobei *mtnKA* kein durchgängiges Transkript zeigt.

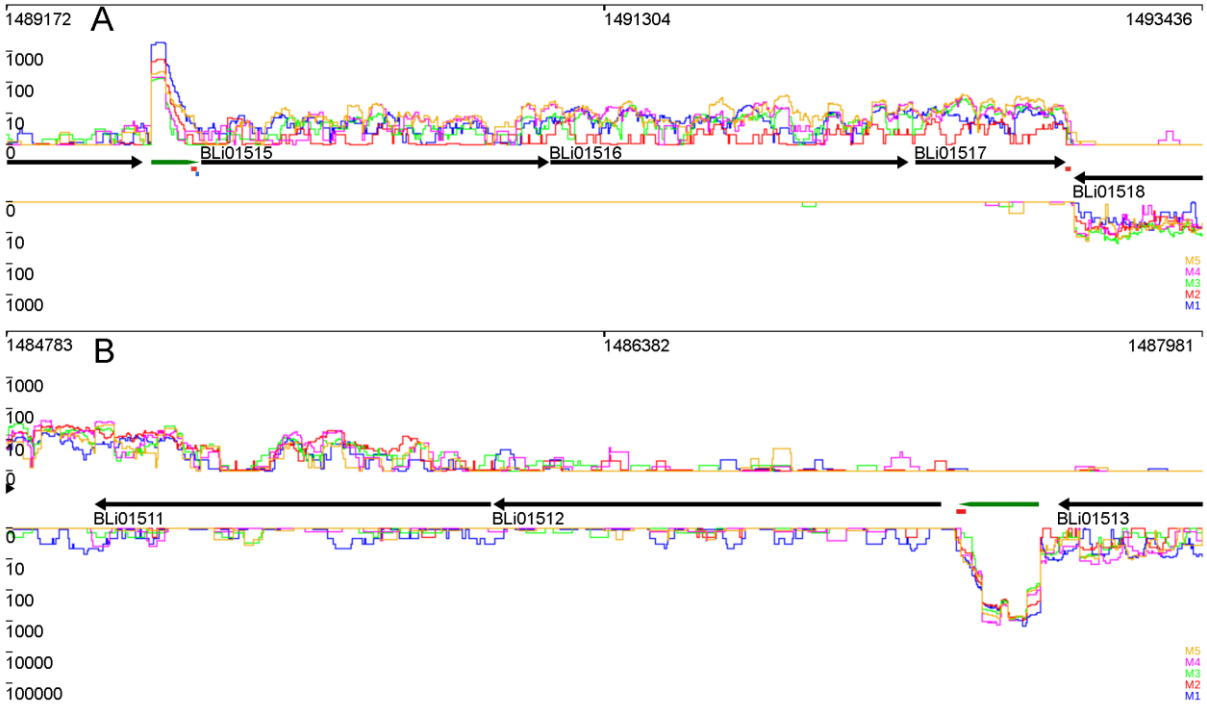

Abb. 27: Transkriptionale Aktivitäten der SAM-*riboswitches* für Gene der SAM-Wiederverwertung im genomischen Kontext
Grafik A zeigt den Kontext für das *mtnWBD* Operon. Grafik B zeigt den Kontext für das *mtnKA* Operon. Rote Kästchen markieren von TransTermHP vorhergesagte Terminatoren, blaue Kästchen markieren manuell vorhergesagte Shine-Dalgarno Sequenzen

Abb. 28 zeigt die Kontexte und Expressionsprofile der *cysH1P1/sat/cysC* und *cysG/sirBC* Operons sowie deren *upstream* Regionen vom Transkriptionsstartpunkt. Das *cysH1P1/sat/cysC* Operon besitzt zwei gut konservierte SigmaA -10 und -35 Promotorboxen (TGcTAAAAT und TTGACT). Beim *cysG/sirBC* Operon ist nur eine -10 Promotorbox zu erkennen wobei diese zwei verschachtelte Patterns zeigt (TGaTTTATA und TATAAT) welche als SigmaA -10 Promotorbox in Frage kommen. Beide Operons zeigen Aktivität wobei auch hier die M1 Phase die höchste Expressionsstärke aufweist. Auffällig ist, dass in der M1 Phase das Transkript von *cysH1P1/sat/cysC* das *cysG/sirBC* Operon miteinschließt, in den anderen Phasen deutet sich jedoch ein Abbruch der Transkription hinter *cysC* und ein neuer Transkriptionsstart für *cysG/sirBC* an. Dies entspricht nicht dem Expressionsverhalten dieses Operons in *B. subtilis* wie von Mansilla et *al.* (Mansilla *et al.*, 2000) beschrieben. Bei *B. subtilis* ist das *cysG* Gen stets Bestandteil des *cysH1P1/sat/cysC* Operons während *sirBC* über die Termination der Transkription der vorhergehenden Gene kontrolliert wird.

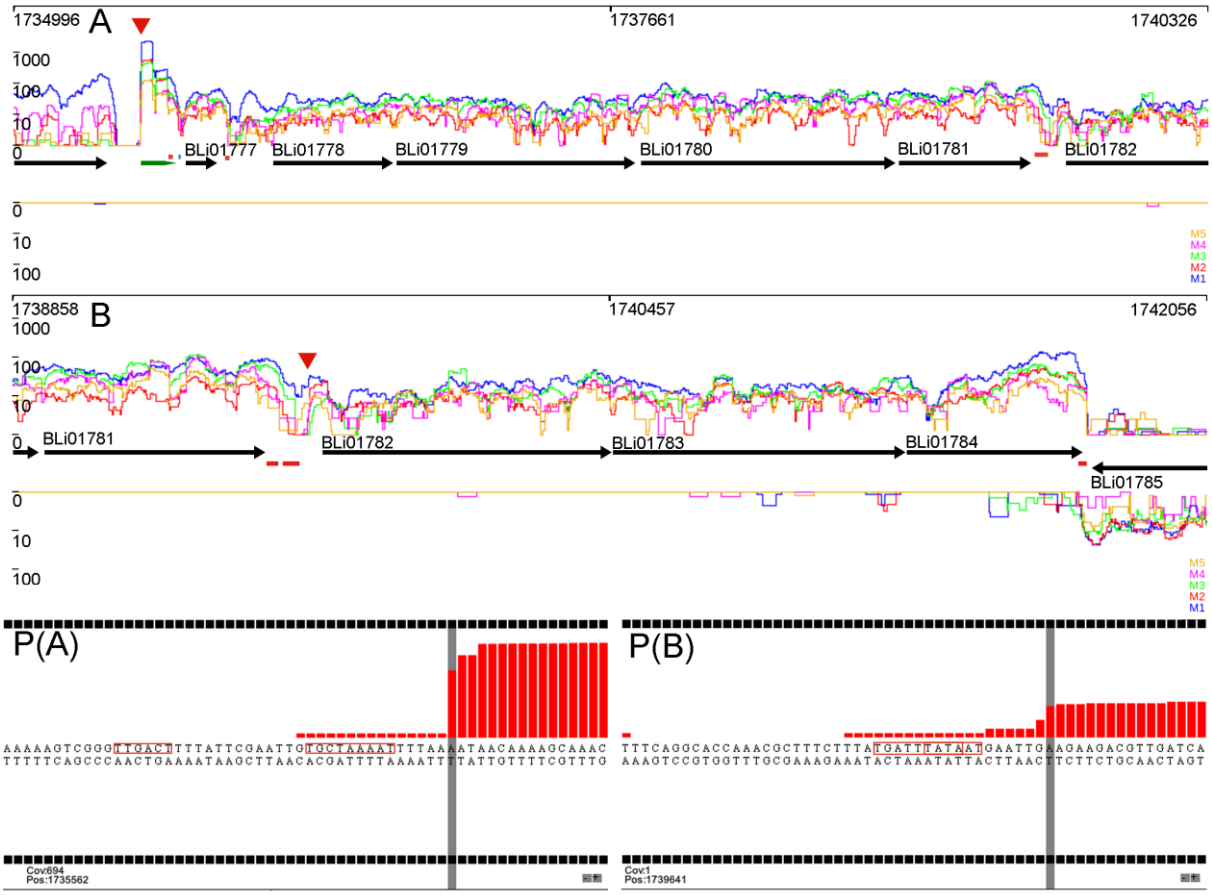

Abb. 28 : Transkriptionale Aktivitäten der *cysH1P1/sat/cysC* und des *cysG/sirBC* Operons im genomischen Kontext
Grafik A zeigt das *cysH1P1/sat/cysC* Operon, der vorhergesagte *riboswitch* ist mit einem grünen Pfeil markiert.Grafik B zeigt das *cysG/sirBC* Operon. Grafik P(A) zeigt die *upstream* Region des *cysH1P1/sat/cysC* Operons in Phase M2. Grafik P(B) zeigt die *upstream* Region des *cysG/sirBC* Operons in Phase M2. Die mit grauen Balken markierten Positionen sind in den Grafiken A und B mit roten Dreiecken markiert. Rote Kästchen markieren von TransTermHP vorhergesagte Terminatoren.
In Grafik P(A) und P(B) sind putative Promotorboxen markiert. Vor dem *cysH1P1/sat/cysC* Operon sind gut konservierte SigmaA -10 und-35 Promotorboxen zu erkennen, vor dem *cysG/sirBC* Operon sind zwei mögliche Promotorstrukturen erkennbar, eine gut konservierte TATAAT Box und eine TG erweiterte TTTATA Box

Bei allen SAM-*riboswitches* konnten im Endbereich Terminatorstrukturen vorhergesagt werden, welche ein weiteres Indiz für die Anwesenheit der vorhergesagten *riboswitches* darstellen (Nudler and Mironov, 2004).

In den Abb. 29, Abb. 30, Abb. 31 und Abb. 32 werden die vorhergesagten *riboswitches* visualisiert. Variationen zum Konsensusmodell sind in den Grafiken gelb markiert und geschehen vor allem in den *loops* an Position 52 und 90. Besonders der *loop* ab Position 52 zeigt große Variabilität in seiner Sequenz und damit auch Größe.

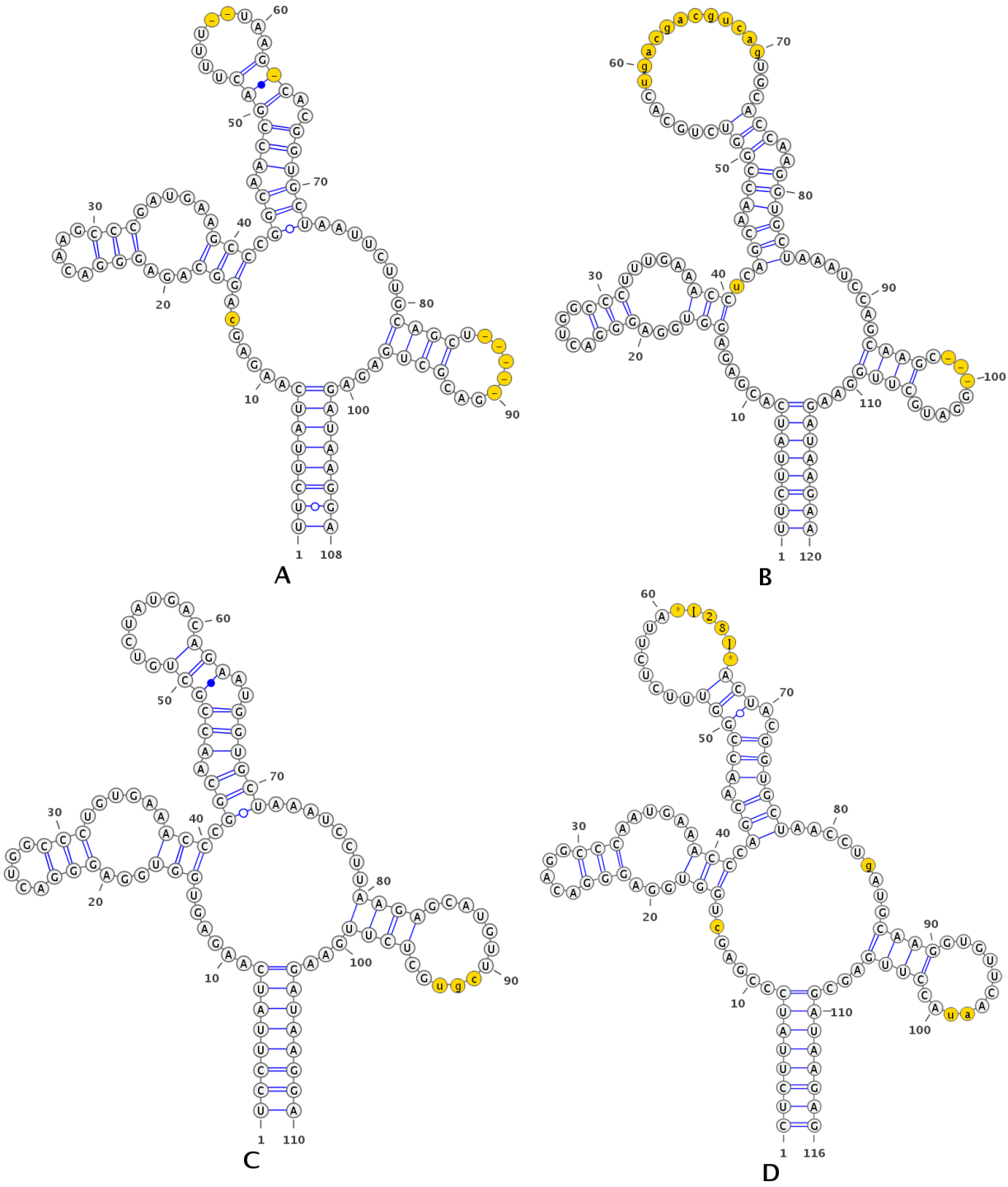

Abb. 29: Vergleich der vorhergesagten SAM *riboswitches* in der Methionin- und SAM-Synthese mit dem Rfam Kovarianzmodell mittels VARNA
Grafik A zeigt den SAM *riboswitch* vor dem *yitJ/metH* Operon. Grafik B zeigt den SAM *riboswitch* vor dem *metIC* Operon. Grafik C zeigt den SAM *riboswitch* vor dem *yxjG* Gen. Grafik D zeigt de n SAM *riboswitch* vor dem *metK* Gen. Unterschiede zum Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell sind durch kleingeschriebene Basen oder durch Bereichsangaben in der Form *[10]* (in diesem Beispiel 10 Basen extra) angegeben. Basenpaarungen sind durch Linien repräsentiert, unpassende Paarungen durch Kreise

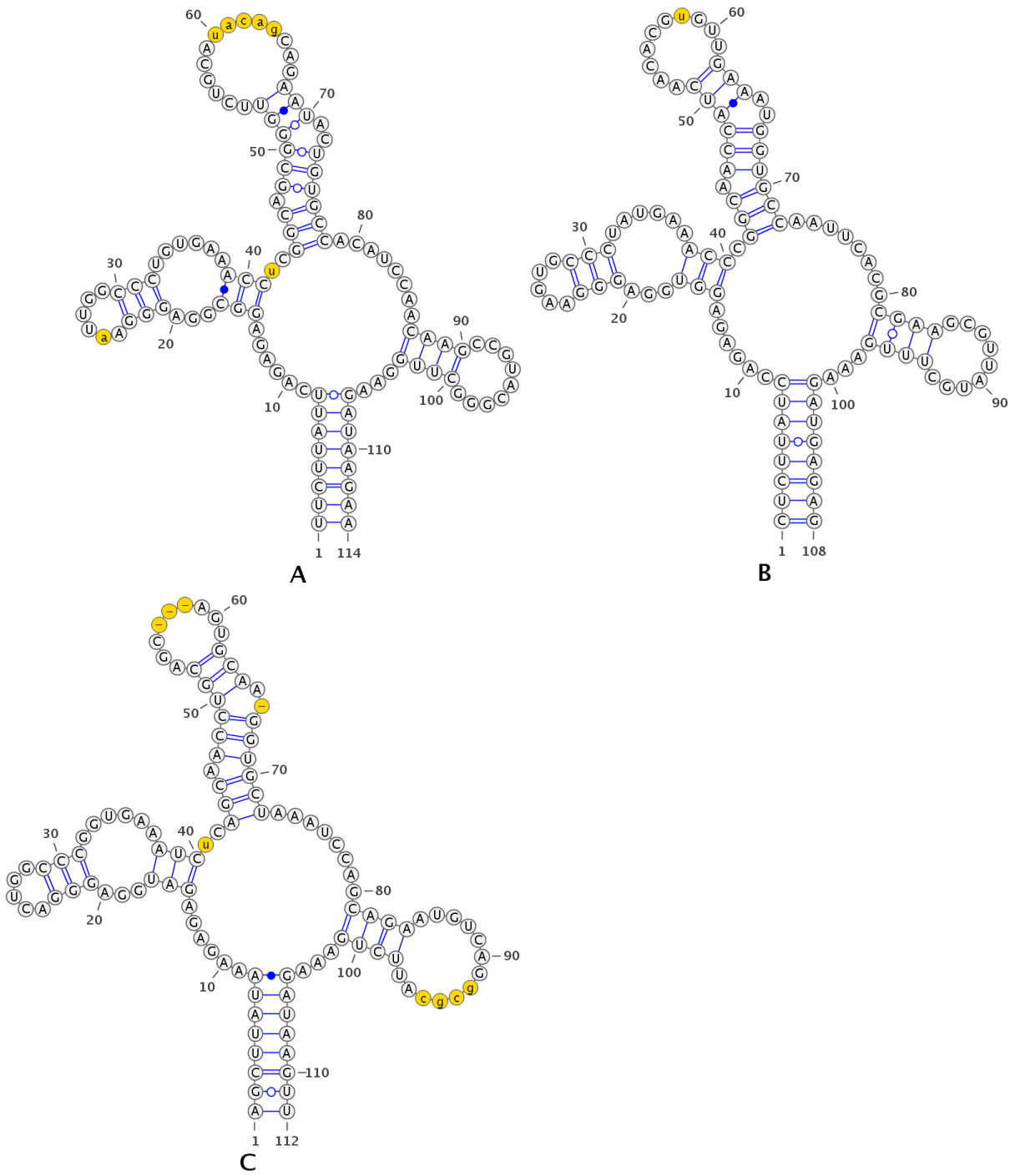

Abb. 30: Vergleich der vorhergesagten SAM *riboswitches* vor Methionintransportergenen aus unseren Daten mit den Rfam Kovarianzmodellen mittels VARNA
Grafik A zeigt den SAM *riboswitch* vor dem *metQ1N1P1* Operon. Grafik B zeigt den SAM *riboswitch* vor dem *metQ2N2P2* Operon. Grafik C zeigt den SAM *riboswitch* vor dem BLi03178 Gen. Unterschiede zum Rfam Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell sind durch kleingeschriebene Basen oder durch Bereichsangaben in der Form *[10]* (in diesem Beispiel 10 Basen extra) angegeben. Basenpaarungen sind durch Linien repräsentiert, unpassende Paarungen durch Kreise

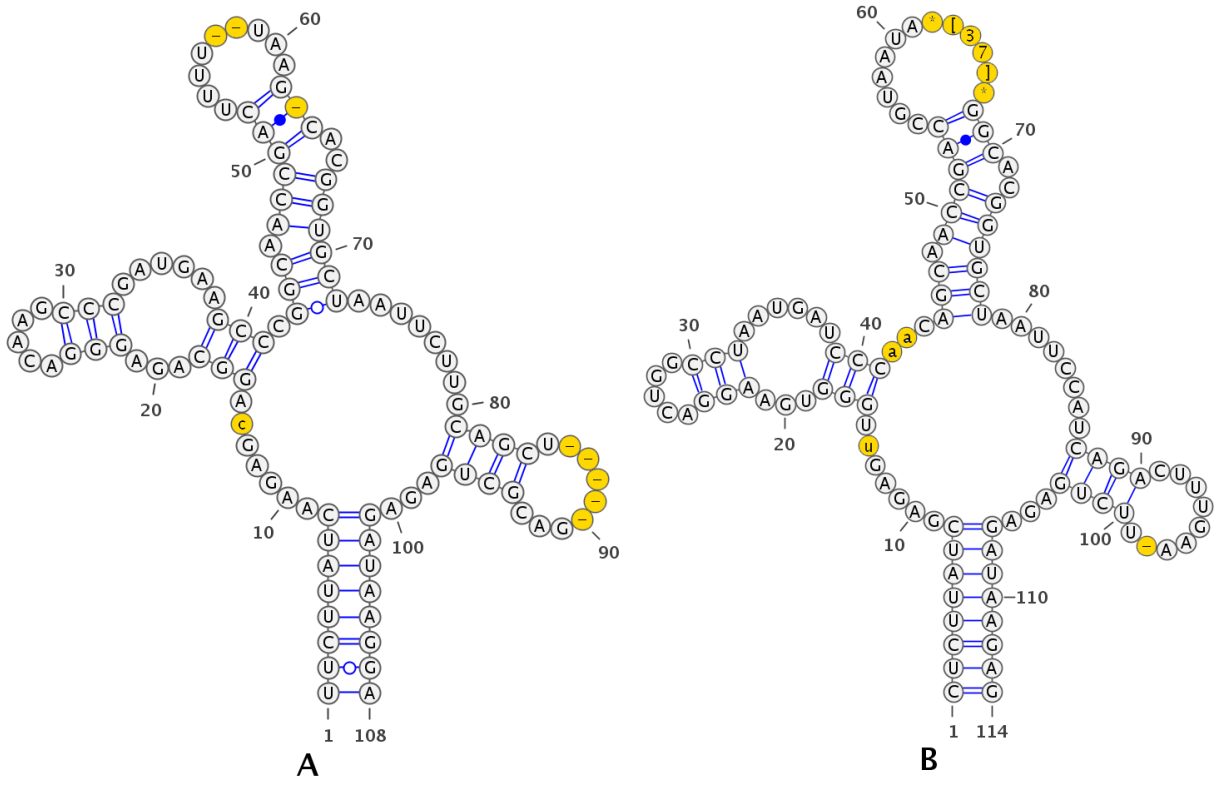

Abb. 31: Vergleich der vorhergesagten SAM *riboswitches* vor Genen der SAM-Wiederverwertung mit dem Rfam Kovarianzmodell mittels VARNA
Grafik A zeigt den SAM *riboswitch* vor dem *mtnWBD* Operon. Grafik B zeigt den SAM *riboswitch* vor dem *mtnKA* Operon. Unterschiede zum Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell sind durch kleingeschriebene Basen oder durch Bereichsangaben in der Form *[10]* (in diesem Beispiel 10 Basen extra) angegeben. Basenpaarungen sind durch Linien repräsentiert, unpassende Paarungen durch Kreise

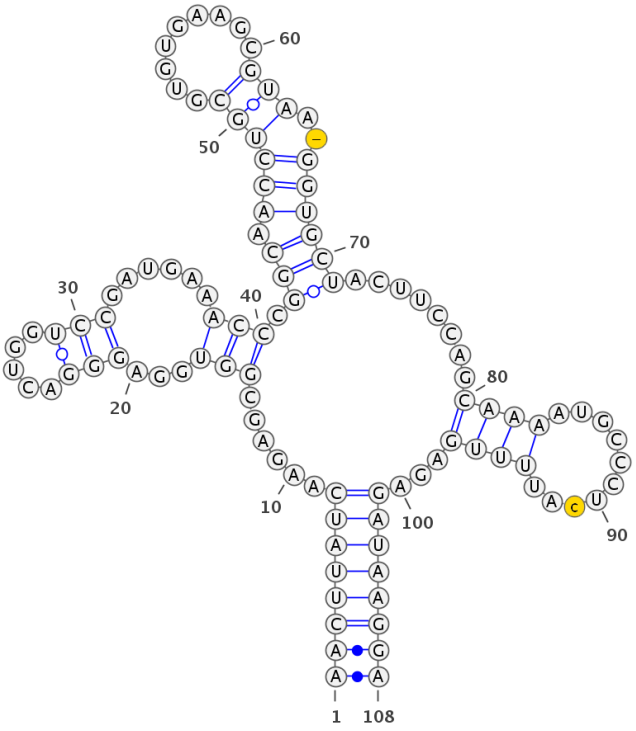

Abb. 32: Vergleich der vorhergesagten SAM *riboswitches* vor dem *cysH1P1/sat/cysC* Operon mit dem Rfam Kovarianzmodell mittels VARNA
Unterschiede zum Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell sind durch kleingeschriebene Basen oder durch Bereichsangaben in der Form *[10]* (in diesem Beispiel 10 Basen extra) angegeben. Basenpaarungen sind durch Linien repräsentiert, unpassende Paarungen durch Kreise

Interessant ist die Variabilität der Sequenz im zweiten *stemloop* (dem Position 52 *loop*) in fast allen SAM-*riboswitches*. Diese Variationen reichen von der Deletion von einigen Basen bis hin zur Insertion von bis zu 37 Basen wie im Beispiel vom *mtnKA* SAM-*riboswitch* oder 28 Basen im *metK* SAM-*riboswitch.* Der dritte *stemloop* (*loop* an Position 90-100) zeigt ebenfalls Variabilität, jedoch nicht im Ausmaß wie der zweite *stemloop.* Die größte Variation ist in den *mtnWBD* und *yitJ/metH* SAM-*riboswitches* zu finden, wo fünf Basen deletiert wurden. Im BLi03178 SAM-*riboswitch* dagegen gibt es eine Insertion von vier Basen.

6.3 *Flavin mononucleotide riboswitches* (FMN-*riboswitches*)

Riboflavin, auch Vitamin B2 genannt, ist eine essentielle Verbindung, die in vielen Stoffwechselreaktionen eine Rolle spielt. Es hat zwei aktive Formen, *flavin adenine dinucleotide* (FAD) und *flavin mononucleotide* (FMN) (Sonenshein *et al.*, 2002).

In *Bacillus licheniformis* DSM13 gibt es ein Operon (*ribTHAED*) für Synthesegene und einen Transporter für *riboflavin* (*ribU*) welche unter der Kontrolle von FMN-*riboswitches* stehen (Vitreschak *et al.*, 2002). In Tabelle 21 sind die Vorhersagen durch die Rfam Analysen aufgelistet. Der *trusted-cutoff* für das FMN Kovarianzmodell liegt bei 32,0. Die Kontexte der beiden identifizierten *riboswitches* sind in Abb. 33 visualisiert.

Tabelle 21: Vorhersagen für FMN-*riboswitches* mittels Rfam Kovarianzmodellen

Phase	Name des Kandidaten	Koordinaten im Genom	Score
1	>UTR5_786	2.381.914-2.382.222	129,19
	>UTR5_796	2.409.010-2.409.325	107,94
	>UTR3_250	848.304+850.082	15,57
2	>UTR5_783	2.381.914-2.382.222	124,19
	>UTR5_792	2.409.010-2.409.334	107,94
	>UTR3_257	848.304+849.599	15,57
3	>UTR5_748	2.381.914-2.382.222	124,19
	>UTR5_755	2.409.010-2.409.312	107,94
	>UTR3_261	848.304+849.198	15,57
4	>UTR5_465	1.284.243+1.284.825	9,07
	>UTR5_870	2.381.914-2.382.265	124,19
	>UTR5_878	2.409.010-2.409.290	107,94
	>UTR3_267	848.304+849.181	15,57
5	>UTR5_839	2.381.914-2.382.227	124,19
	>UTR5_848	2.409.010-2.409.314	107,94
	>sRNA_350	2.997.746+2.997.966	14,30

Anhand von der NPKM-Werte Tabelle 22 lässt sich zeigen, dass die FMN-*riboswitches* stärker transkribiert werden als die Operons, vor denen sie lokalisiert sind. Anhand der Ratios sieht man, dass *ribU* und *ribTHAED* in unterschiedlichen Phasen regulieren. Der *ribU*

riboswitch scheint in den Phasen 2 und 3 am wenigsten zu reprimieren während der *ribTHAED riboswitch* während der Phase 4 am wenigsten reprimiert. Die Replikate zeigen bei diesen Operons starke Schwankungen (siehe P15_FMNs.tsv im Anhang unter NPKMListen) in den Expressionsmustern. Zwei Replikate (M und L) zeigen die höchste Aktivität von *ribU* in Phase 2 und 3 und *ribTHAED* in Phase 4. Das R Replikat dagegen zeigt für *ribU* und *ribTHAED* die höchste Aktivität in Phase 2 wobei auch hier Phase 3 für *ribU* und Phase 4 für *ribTHAED* hohe Aktivitäten aufweisen. Möglicherweise sind diese Schwankungen der Aktivitäten auf zu große Unterschiede in den Probezeitpunkten zurückzuführen.

Tabelle 22: NPKM Werte der FMN *riboswitches* und deren regulierten Operons in den Phasen M1-M5

riboswitch	**M1 NPKM**	**M2 NPKM**	**M3 NPKM**	**M4 NPKM**	**M5 NPKM**
ribU	2.895	2.556	2.224	2.436	2.223
ribTHAED	2.222	2.660	1.636	1.815	2.041
Operon	**M1 NPKM**	**M2 NPKM**	**M3 NPKM**	**M4 NPKM**	**M5 NPKM**
ribU	62	188	160	60	55
ribTHAED	58	127	65	315	100

Tabelle 23: Verhältnisse der FMN *riboswitches* zu den regulierten Operons in den Phasen M1-M5

riboswitch	**M1 Ratio**	**M2 Ratio**	**M3 Ratio**	**M4 Ratio**	**M5 Ratio**
ribU	46,69	13,59	13,9	40,6	40,42
ribTHAED	38,31	20,94	25,17	5,76	20,41

In der Abb. 33 sind die Kontexte der FMN-*riboswitch* kontrollierten Operons visualisiert. Das *ribU* Gen zeigt die die höchste Aktivität in Phase M2 und M3. Das *ribTHAED* Operon zeigt ein differentielles Expressionsverhalten der einzelnen Gene im Operon. Das *ribT* Gen (BLi02471) zeigt in allen Phasen im Vergleich zu den restlichen Genen im Operon eine erhöhte Aktivität. Zusätzlich deutet sich vor dem Gen ein alternativer Promotor oder eine regulatorische Struktur an. Das Gen *ribH* (BLi02472) wird in den Phasen M4 und M5 ebenfalls verstärkt exprimiert. Auch hier deutet sich in M4 und M5 ein alternativer Promotor oder eine regulatorische Struktur an. Die restlichen Gene zeigen die größte transkriptionelle Aktivität in Phase M2 und M3 während sie in M1 sowie M4 und M5 nur geringe transkriptionelle Aktivität aufweisen. Tabelle 24 zeigt die NPKM Werte bei denen die Gene *ribAED, ribH* und *ribT* als eigenständige Transkripte behandelt werden. Diese Aufteilung des Operons reflektiert das in Abb. 33 gezeigte Expressionsverhalten besser als die Betrachtung

von *ribTHAED* als geschlossenes Operon, da hier die Zunahme der Expressionsaktivität von *ribT* und *ribH* in Phase M4 und M5 eindeutiger gezeigt wird.

Die Unterschiede in den Replikaten machen eine genaue Bestimmung des Expressionsverhaltens schwierig. Als Gegenbeispiel zu den Replikaten M1-M5 wird in Abb. 34 das Expressionsverhalten des *ribTHAED* Operons in den Replikaten R1-R5 dargestellt. Auffällig ist in diesem Beispiel, dass *ribT* und *ribH* in Phase R2 nicht die typischen Muster für die Aktivität von Promotoren oder regulatorischen Strukturen zeigen. Abb. 35 zeigt die *upstream*-Regionen dieser beiden Gene. Vor *ribT* ist eine SigA -10 Promotorbox zu erkennen wogegen *ribH* keine bekannten Promotorpatterns aufweist.

Tabelle 24: NPKM Werte der möglichen *ribT*, *ribH* und *ribAED* Transkripte

Operon	M1 NPKM	M2 NPKM	M3 NPKM	M4 NPKM	M5 NPKM
ribAED	27	97	43	25	28
ribH	43	122	45	1.164	289
ribT	279	324	234	1.240	351

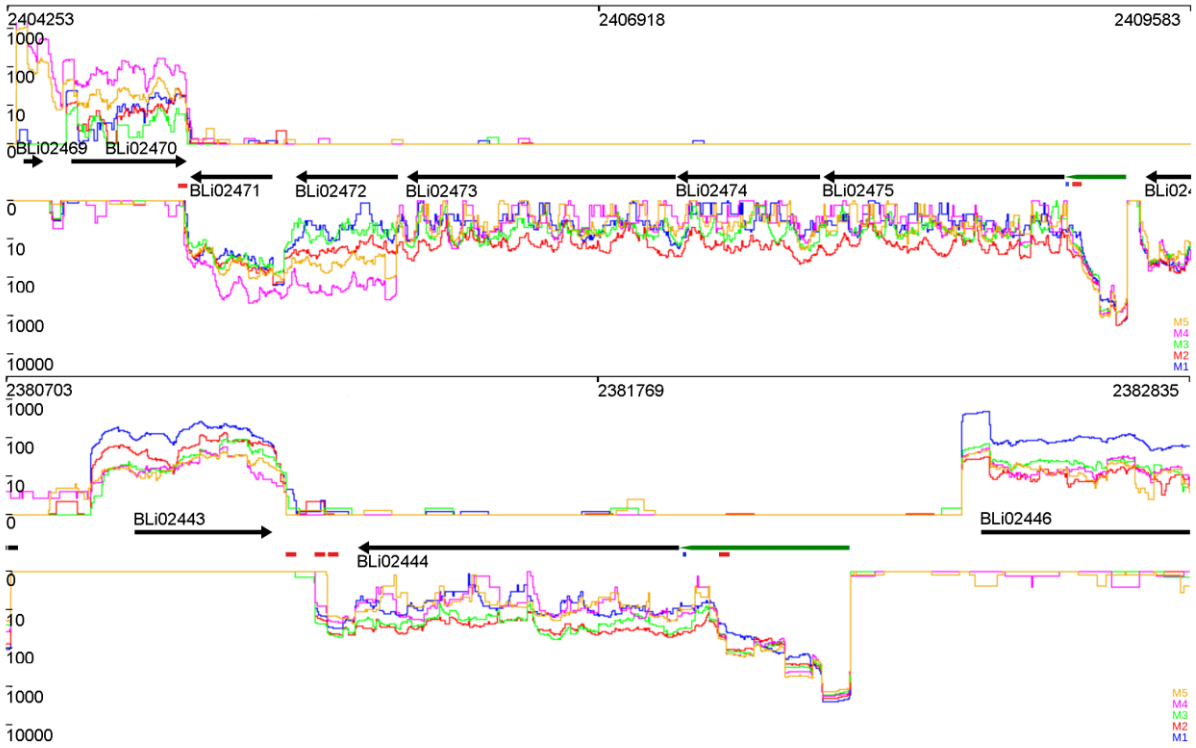

Abb. 33: Transkriptionale Aktivitäten der vorhergesagten FMN *riboswitches* im genomischen Kontext in den Phasen M1-M5
Grafik A zeigt das Riboflavinsynthese-Operon *ribTHAED*(BLi02471-BLi02475). Grafik B zeigt das Riboflavintransporter Gen *ribU* (BLi02444). Die Bereiche mit den vorhergesagten *riboswitches* sind durch grüne Pfeile markiert. Rote Kästchen markieren mit TransTermHP vorhergesagte Terminatoren, blaue Kästchen markieren manuell vorhergesagte Shine-Dalgarno Sequenzen

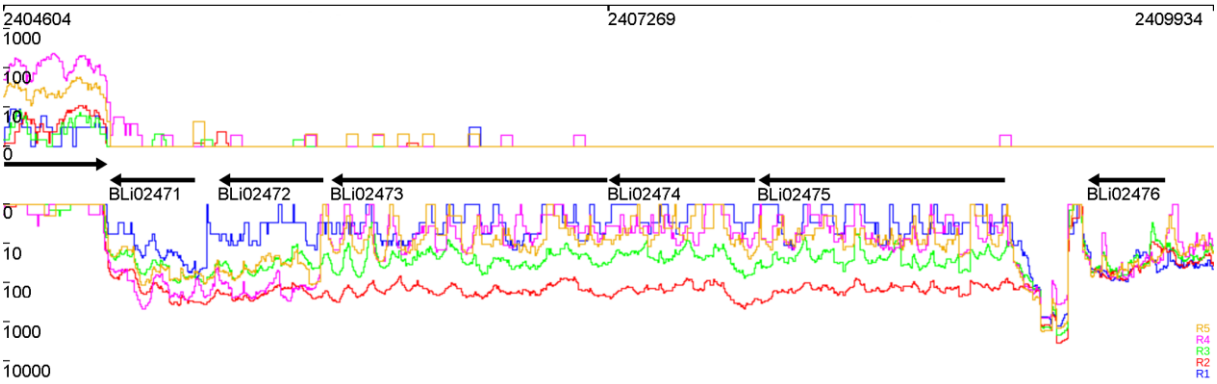

Abb. 34: Expressionsprofil des *ribTHAED* Operons in den Phasen R1-R5.
Innerhalb von Phase R2 sind für *ribH* und *ribT* keine seperaten Transkriptionsstarts zu erkennen wie es in R4 und R5 der Fall ist

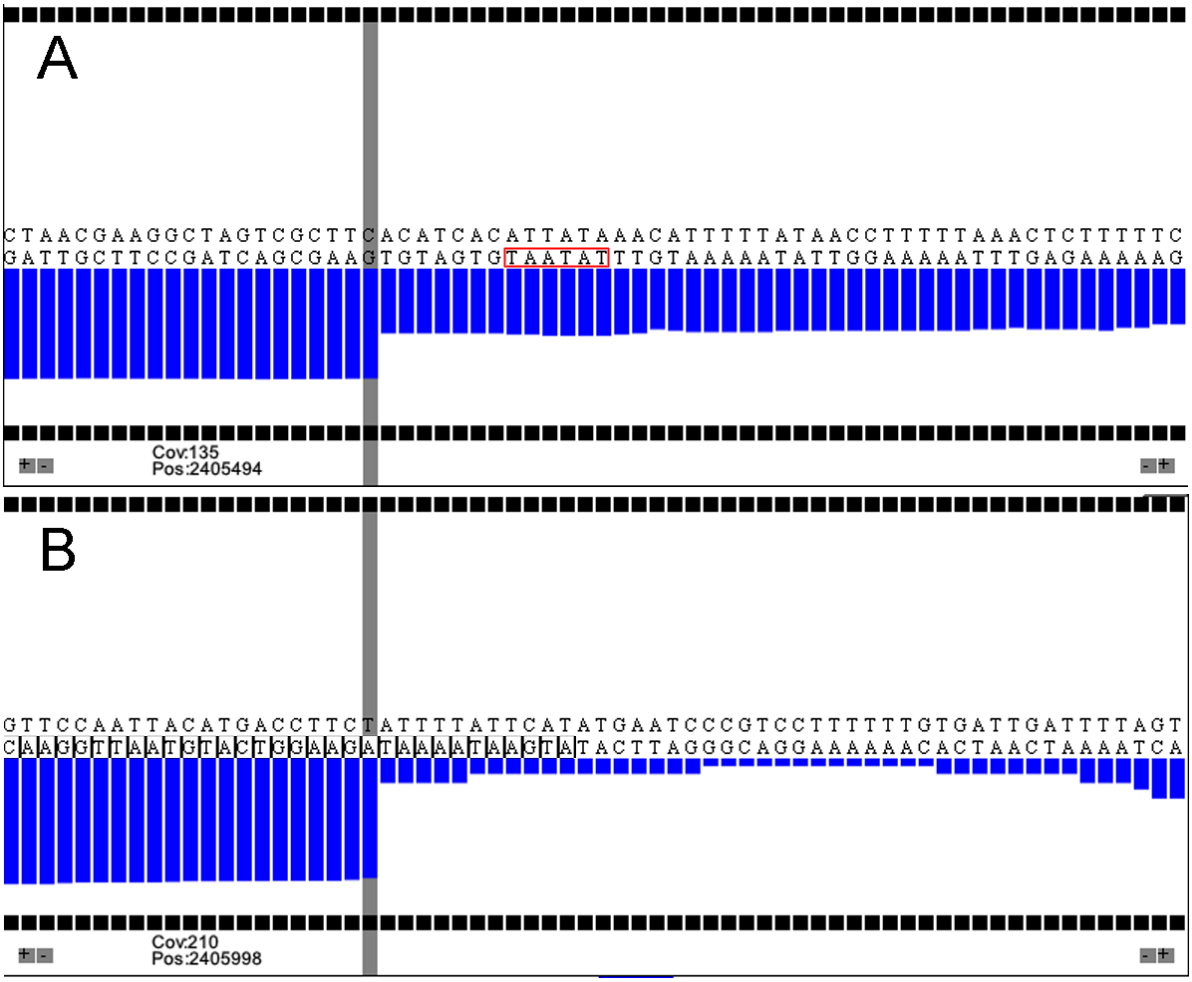

Abb. 35: Putative Transkriptionsstarts *upstream* von *ribT* (A) und innerhalb von *ribH* (B)
Bei *ribT* ist eine SigA -10 Promotorbox zu erkennen (rot umrandet) wogegen der putative Transkriptionsstart in *ribH* kein bekanntes *pattern* aufweist

In Abb. 36 sind die gefundenen FMN-*riboswitches* mittels VARNA visualisiert.

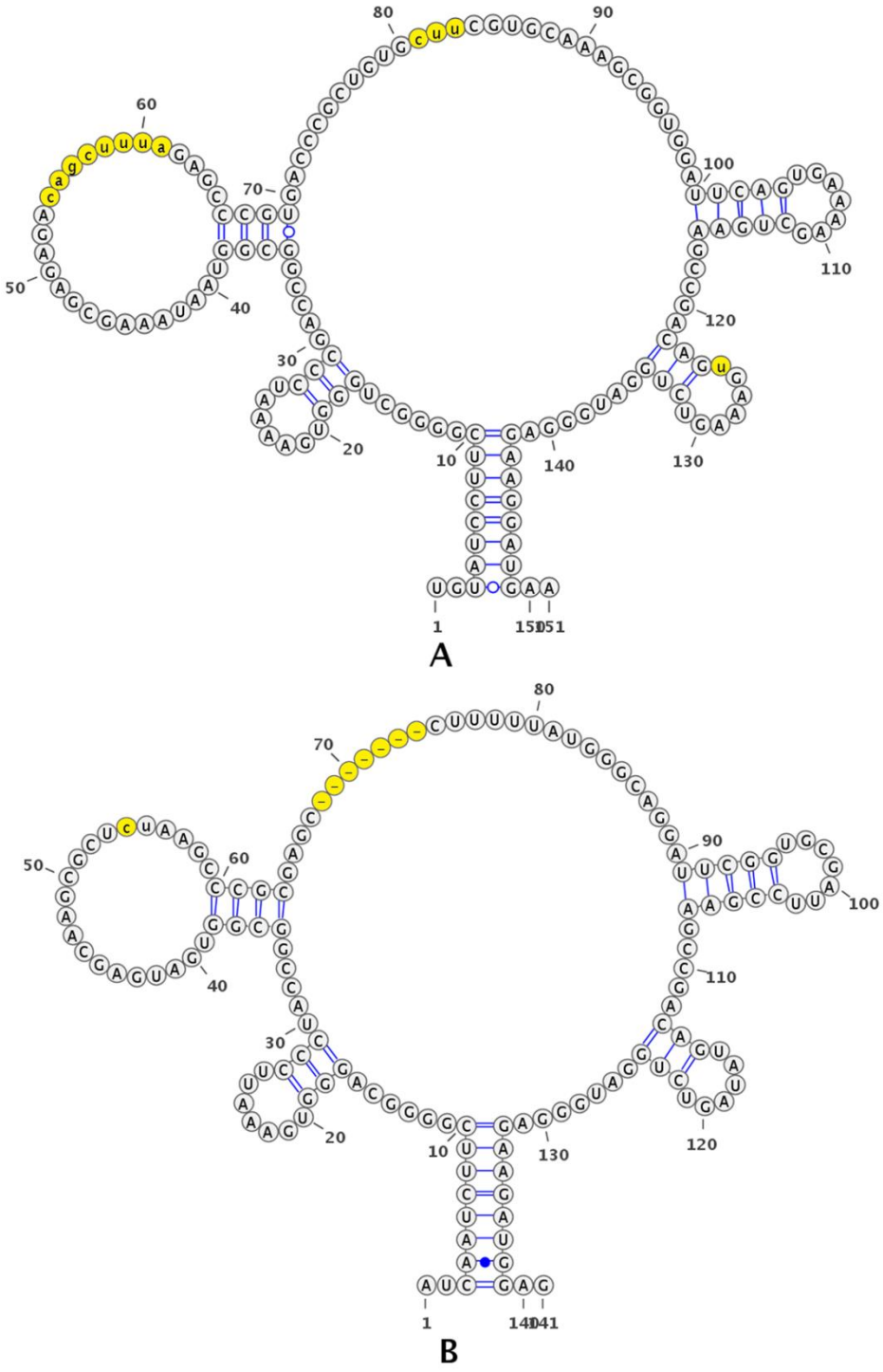

Abb. 36: Vergleich der vorhergesagten FMN *riboswitches* mit den Rfam Kovarianzmodell mittels VARNA
Grafik A zeigt den *riboswitch* vor dem *ribTHAED* Operon. Grafik B zeigt den *riboswitch* vor dem *ribU* Gen. Unterschiede zum Kovarianzmodell sind gelb markiert. Fehlende Basen im Vergleich zum Modell sind durch – markiert. Insertionen im Vergleich zum Modell sind durch kleingeschriebene Basen oder durch Bereichsangaben in der Form *[10]* (in diesem Beispiel 10 Basen extra) angegeben. Basenpaarungen sind durch Linien repräsentiert, unpassende Paarungen durch Kreise

Variationen sind vor allem im zweiten und im zentralen *loop* zu finden. Die Variation im *ribTHAED riboswitch* führt zu einer Vergrößerung im zweiten *stemloop*, die Variation im *ribU riboswitch* führt zu einem verkleinerten zentralen *stemloop.*

6.4 Response regulator aspartate phosphatases (Rap Gene)

Die *response regulator aspartate phosphateses,* kurz Rap, sind eine Gruppe von Phosphatasen die im *quorom sensing* beteiligt sind (Perego, 2013). Funktional haben sie Einfluss auf die Kompetenz, Sporulation und Biofilmbildung und bewirken im aktiven Zustand eine Unterdrückung dieser indem sie die Phosphorilierungskaskaden zur Aktivierung der für diese Prozesse spezifischen Regulatoren unterbrechen. Reguliert werden die Rap Proteine durch kleine Signalpeptide, genannt Phr, welche bei Bindung eine Konformationsänderung in den Rap Proteinen bewirken und diese inaktivieren. Die Phr-Peptide werden von den Zellen exportiert und während dieses Exports modifiziert um die aktive Variante des Peptids zu generieren. Diese aktivierten, extrazellulären Phr Peptide werden dann in die Zelle importiert und bewirken damit die Regulation der Rap Proteine (Perego and Hoch, 1996).

Insgesamt sind in *B. licheniformis* DSM13 sieben Rap Gene annotiert basierend auf der Homologie zu *Bacillus subtilis* 168. Jarmer *et al.*haben für die Gene Rap bioinformatische Promotorvorhersagen gemacht und dabei gezeigt, dass die Rap Gene in *Bacillus subtilis* durch SigA reguliert werden (Jarmer *et al.*, 2001).

- *rapA,* wobei es zwei Varianten, *rapA1* und *rapA2* gibt. In *B. subtilis* 168 wird *rapA* durch SigA kontrolliert.
- *rapG,*in *B. subtilis* 168 kontrolliert durch SigA.
- *rapH,* in *B. subtilis* 168 kontrolliert durch SigA.
- *rapI,* in *B. subtilis* 168 kontrolliert durch SigA.
- *rapD,* in *B. subtilis* 168 kontrolliert durch SigA, SigM (Eiamphungporn and Helmann, 2008) und SigX (Huang and Helmann, 1998).
- *rapK,* in *B. subtilis* 168 kontrolliert durch SigA.

Außerdem gibt es Untersuchungen, die zeigen, dass die Phr Peptide durch SigH kontrolliert werden (Mcquade *et al.*, 2001).

Damit sind vor den *rap* und *phr* Genen Variationen der *patterns* (nach Sonenshein *et al.*, 2002) aus Tabelle 25 zu erwarten.

Tabelle 25: Erwartete *patterns* vor den Transkriptionsstartpunkten der *rap* und *phr* Gene

Pattern	**Konsensussequenz**
SigA	**TTG**ACA-N15,17-**TA**TAA**T-**N6
SigH	**AGGA**NN**T**-N13,15-**GAA**T-N9
SigM	**GAAAAC**-N17-**CG**TC**-**N9
SigX	**TGTAAC-**N17-**CG**AC**-**N8

Abb. 37 zeigt die Expressionsprofile der beiden *rapA/rapH/phrA* Gene in *Bacillus licheniformis* DSM13 in den Phasen M1-M5.

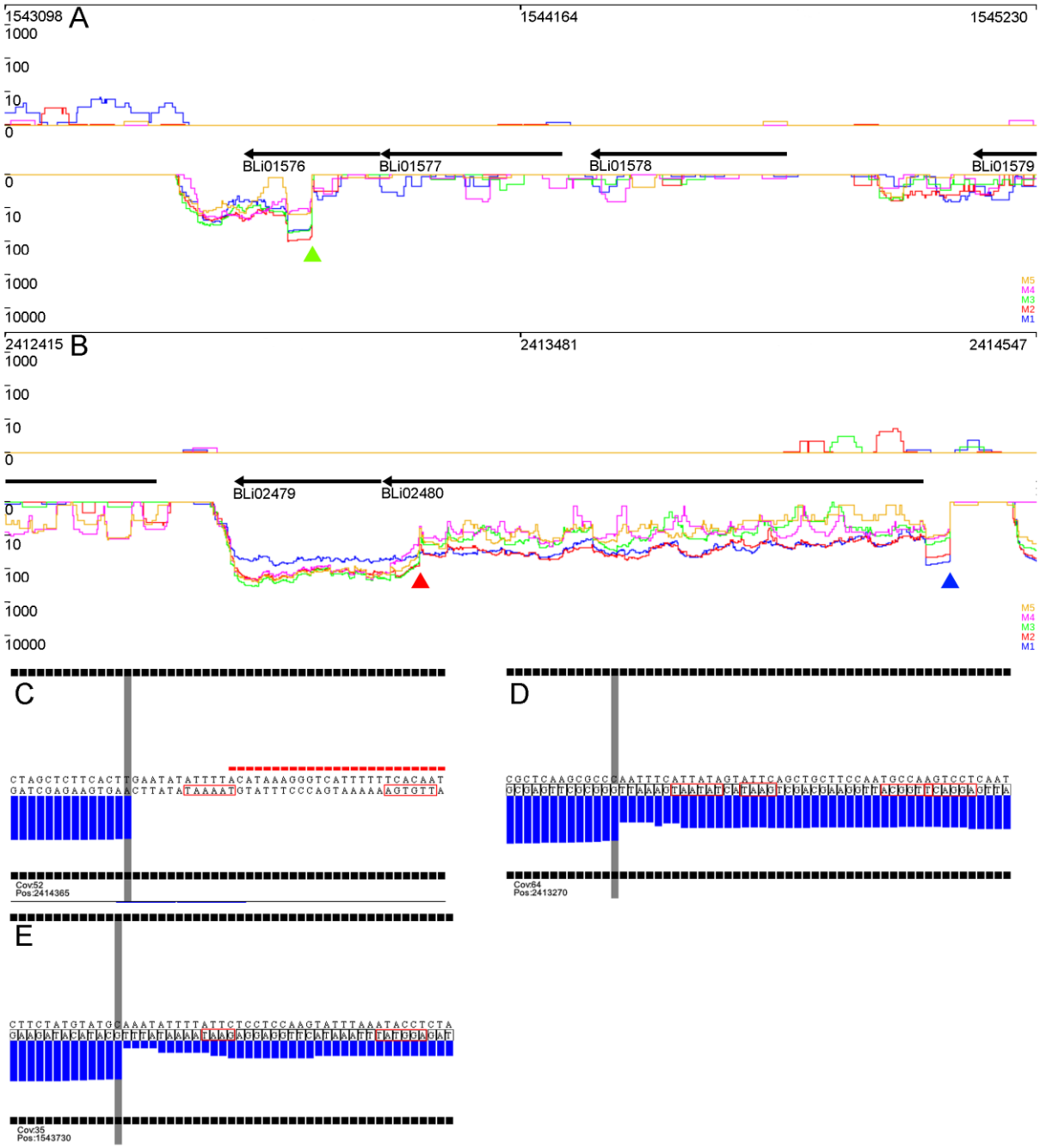

Abb. 37: Expressionsprofile und *upstream* Regionen der (BLi01577/BLi01578), *rapA2*(BLi02480) und *phrA1*(BLi01576) und *phrA2*(BLi02479) Gene

Grafik A zeigt die Expressionsprofile von *rapA1/rapH/phrA1*. Grafik B zeigt die Expressionsprofile von *rapA2/phrA2*. Grafik C zeigt die *upstream* Region vom TSS von *phrA1*. Die entsprechende TSS ist in Grafik A mit einem grünen Pfeil markiert. Grafik D zeigtdie *upstream* Region von *phrA2.* Die entsprechende TSS ist in Grafik B mit einem roten Pfeil markiert. Grafik E zeigt die *upstream* Region von *rapA2*. Die entsprechende TSS ist in Grafik B mit einem blauen Pfeil markiert. In den Grafiken C,D und E sind mögliche *patterns* für Promotoren rot umrandet

Anhand von Abb. 37 lässt sich zeigen, dass *rapA2* und *phrA2* aktiv transkribiert werden. Die Gene *rapA1*, *rapH* sind anscheinend inaktiv, da keine durchgängigen Transkripte vorliegen. Das *phrA1* Gen zeigt jedoch transkriptionelle Aktivität.

In der *upstream* Region von *rapA2* lassen sich SigA -10 und -35 Promotorboxen identifizieren (**TTGTGA-**N17-**TAAAAT**-N6). Das *phrA2* Gen hat in seiner *upstream* Region Kandidaten für SigA (**TTGGCA**-N18-**TATAAT**-N6) und SigH (**AGGACT**-N17-**GAAT**-N14) Promotorboxen. Die Abstände zum Transkriptionsstart deuten aber darauf hin, dass nur der SigA Promotor aktiv ist. Interessant in diesem Fall ist die Aktivität in der Phase M1. In M1 deutet das Expressionsprofil an, dass *phrA2* in dieser Phase keinen eigenen aktiven Promotor hat und mit *rapA2* im Operon von dessen Promotor mitabgelesen wird. In den anderen Phasen zeigt sich ein eigenständiger Transkriptionsstart für *phrA2*. Beim *phrA1* Gen sind Kandidaten für SigH -10 und -35 Promotorboxen zu finden (**AGGTAT**-N16-**GAAT**-N9). Zusätzlich gibt es ein TAAAAT Muster 4 Basen vom Transkriptionsstart entfernt. Dieses Pattern ist aber weder TG erweitert noch ist eine -35 Box für einen SigA Promotor zu erkennen. Es liegt außerdem sehr nahe am Transkriptionsstart. Daher scheint der SigH Promotor der einzige aktive Promotor für dieses Gen zu sein.

Abb. 38 zeigt die Expressionsprofile von *rapG* und *phrG* in den Phasen M1-M5.

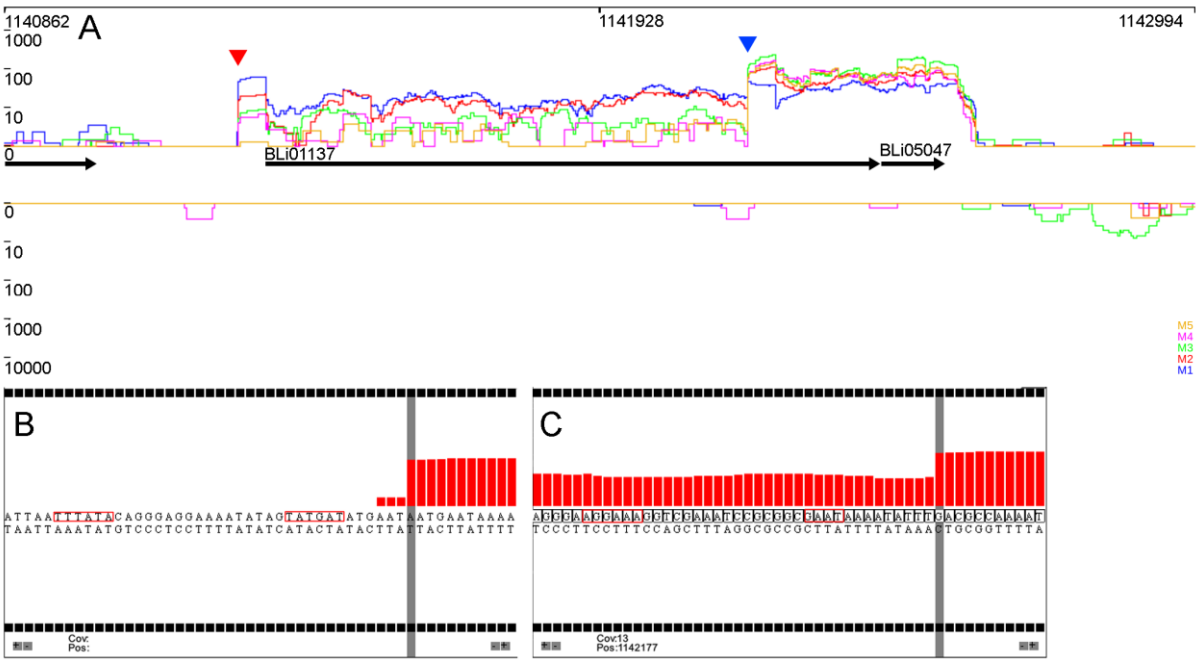

Abb. 38: Expressionsprofile und *upstream* Regionen von *rapG*(BLi01137) und *phrG* (BLi05047)
Grafik A zeigt die Expressionsprofile der Gene in den Phasen M1-M5. Grafik B zeigt die *upstream* Region vor dem *rapG* Gen. Die entsprechende TSS ist in Grafik A mit einem roten Pfeil markiert. Grafik C zeigt die upstream Region vor dem *phrG* Gen. Die entsprechende TSS ist in Grafik A mit einem blauen Pfeil markiert. In den Grafiken B und C sind mögliche *patterns* für Promotoren rot markiert

Anhand der Abb. 38 lässt sich zeigen, dass das Gen *rapG* in den Phasen M1 und M2 am aktivsten ist und die Aktivität in Phase M3 abnimmt. In Phase M4 und M5 scheint es nur noch minimal transkribiert zu werden. Das *phrG* Gen dagegen zeigt in allen Phasen Transkriptionsaktivität wobei Phase M1 etwas weniger aktiv zu sein scheint als M2-M5.

In der *upstream* Region des *rapG* Gens sind mögliche SigA -35 und -10 Promotorboxen zu erkennen (**TTTATC**-N17-**TATGAT**-N6). Die -35 weicht stark vom Konsensus TTGACA ab, insbesondere das Guanin an der dritten Position des *patterns* fehlt. Die Abstände der Boxen zueinander und zum TSS entsprechen dem erwarteten Muster. Im *upstream* Bereich des *phrG* Gens sind die Promotorboxen für SigH (**AGGAAA**-N16-**GAAT**-N9) zu erkennen. Damit wird das *rapG* Gen anscheinend durch SigA und *phrG* durch SigH kontrolliert. Wie auch bei *rapA2/phrA2* scheint in diesem Beispiel die Expression in Phase M1 stark vom Promotor des Gens mitgetragen zu werden. Erst in den späteren Phasen scheint der *phrG* eigene Promotor aktiv zu werden während die Aktivität des *rapG* Promotors abnimmt.

Abb. 39 zeigt die Expressionsprofile der *rapI* und *phrI* Gene in den Phasen M1-M5.

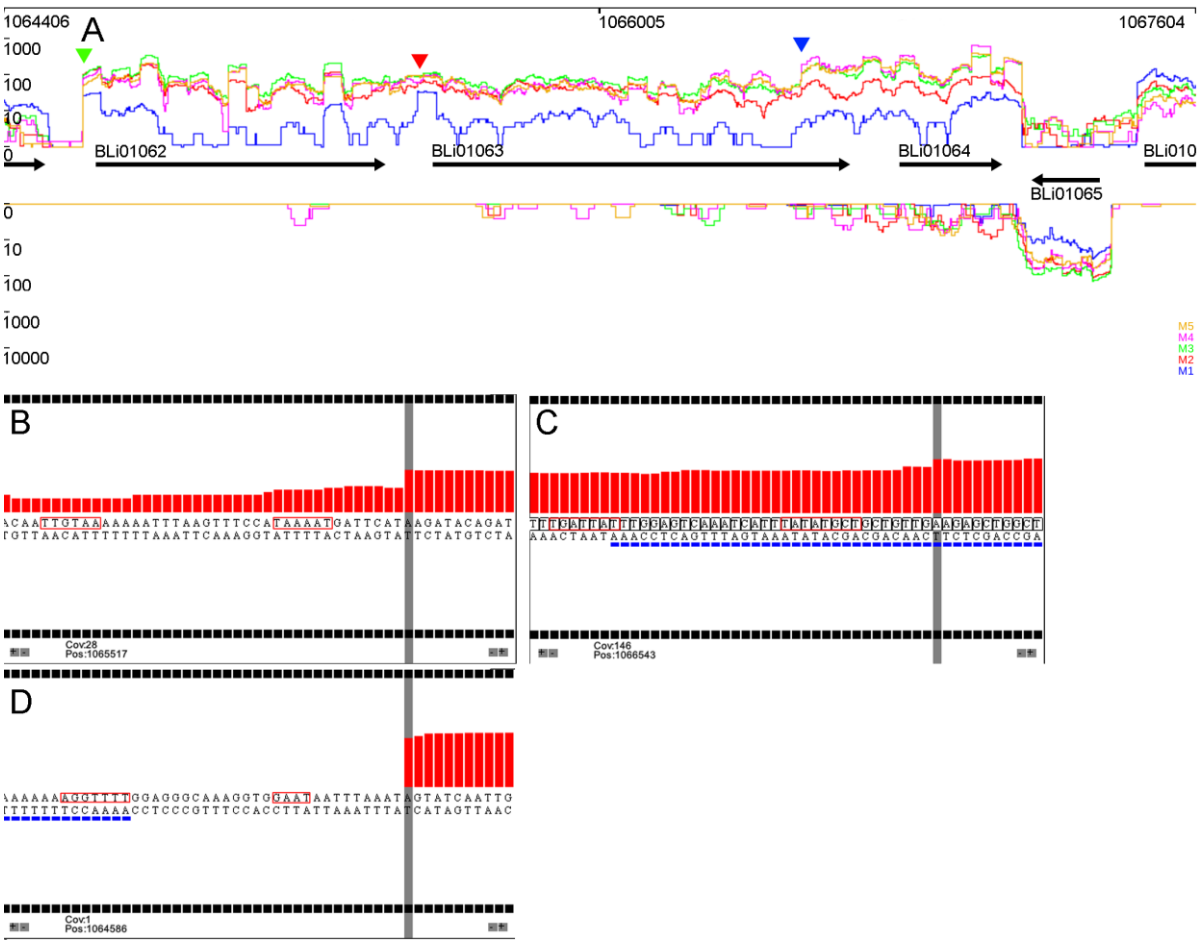

Abb. 39: Expressionsprofile und *upstream* Regionen von *rapI*(BLi01063) und *phrI* (BLi01064)
Grafik A zeigt die Expressionsprofile der Gene. Grafik B zeigt die *upstream* Region vor *rapI*. Die entsprechende TSS ist in Grafik A mit einem roten Pfeil markiert. Grafik C zeigt die *upstream* Region des *phrI* Gens. Die entsprechende TSS ist in Grafik A mit einem blauen Pfeil markiert. Die Grafik D zeigt die *upstream* Region von *yhaR*(BLi01062). Die entsprechende TSS ist in Grafik A mit einem grünen Pfeil markiert. In Grafik B, C und D sind mögliche *patterns* für Promotoren rot markiert

Die Expressionsprofile von *rapI* und *phrI* sind schwierig auszuwerten, da es viele Anstiege in der Basenaktvität gibt, die typisch für Transkriptionsstarts sind. Jene Anstiege, vor denen mögliche Promotoren liegen können, sind in der Grafik A der Abb. 39 mit Pfeilen markiert und deren *upstream* Region ist im Detail dargestellt. Das *rapI* Gen verfügt über einen möglichen SigA Promotor (**TTGTAA**-N17-**TAAAAT**-N7), jedoch scheint dieser nur in der Phase M1 aktiv genutzt zu werden. Während der Phasen M2-M5 gibt es an dieser Stelle keine auffälligen Anstiege. Stattdessen scheint das Transkript des vorhergehenden Gens, *yhaR*, *rapI* mitabzulesen. Das *yhaR* Gen besitzt in seiner *upstream* Region einen SigH Promotor (**AGGTTTT**-N15-**GAAT**-N9) womit, wenn das Transkript tatsächlich *polycistronisch* ist, *rapI* auch unter der Kontrolle von SigH stehen würde.

Das *phrI* Gen besitzt keines der Eingangs erwähnten *pattern*. Stattdessen sind vor einem schwachen Anstieg der Basenaktivität zwei *patterns* (**TGATTAT**-N16-**TATATGCT-**N7) zu

sehen, die denen von SigE Promotoren ähneln (TC**AT**ATT-N15-**CATA**CGA**T**-N6) (Eichenberger *et al.*, 2003). Während der Phasen M1 und M2 zeigt sich an der Position dieses möglichen Promotors noch kein für einen Transkriptionsstart typisches Muster und die Aktivität scheint allein von den Promotoren für die *rapI* oder *yhaR* Gene abzuhängen. Erst während der Phasen M3-M5 zeigt sich ein deutlicher Anstieg in der Aktivität. Während dieser Phasen steigt dann auch die Aktivität im Verhältnis zu *rapI*.

Abb. 40 zeigt die Expressionsprofile der *rapD* und *phrD* Gene in den Phasen M1-M5.

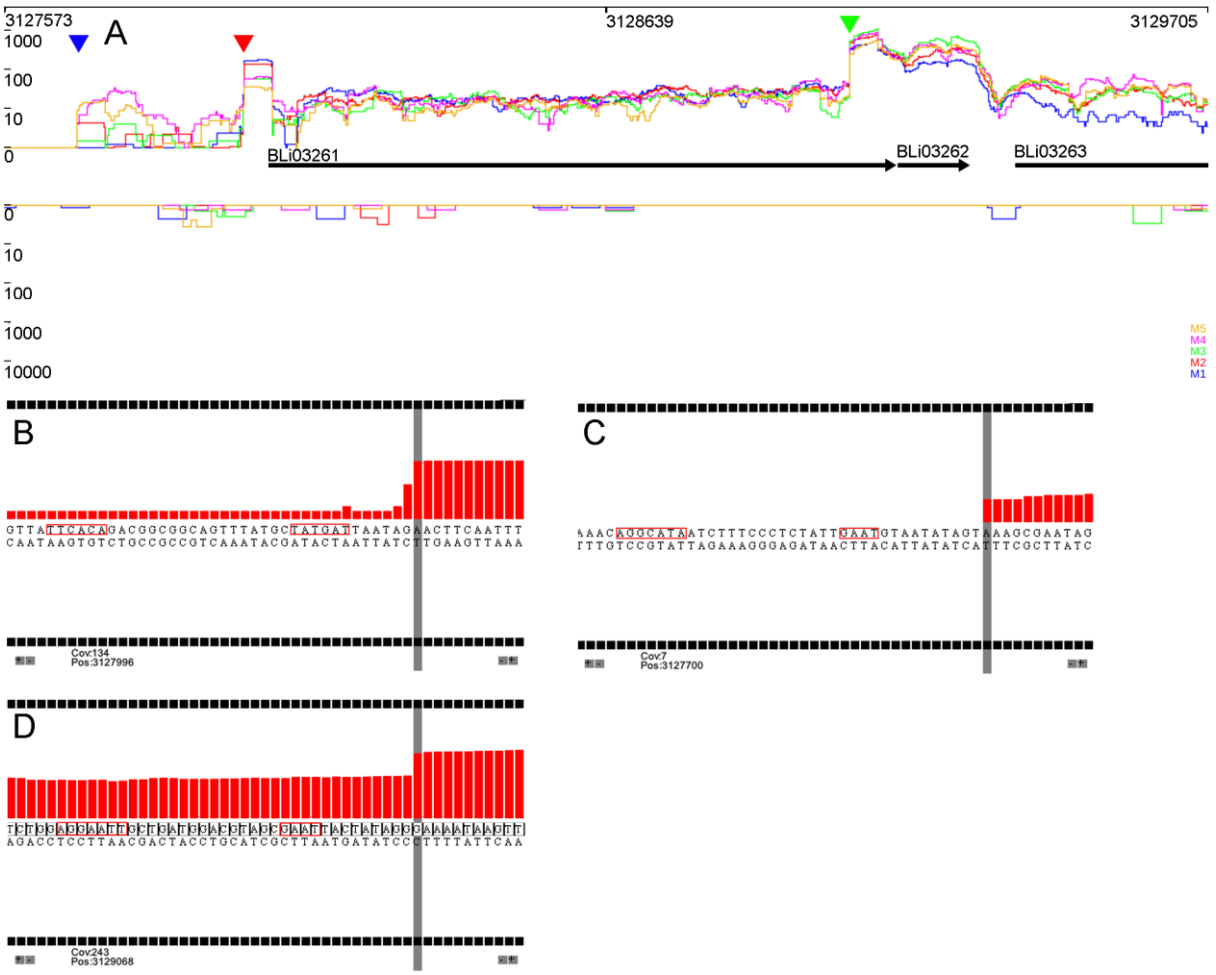

Abb. 40: Expressionsprofile und *upstream* Regionen von *rapD*(BLi03261) und *phrD* (BLi03262)
Grafik A zeigt die Expressionsprofile der Gene. Grafik B zeigt den ersten möglichen Transkriptionsstart *upstream* von *rapD*. Die entsprechende TSS ist in Grafik A mit einem roten Pfeil markiert. Grafik C zeigt den zweiten möglichen Transkriptionsstart *upstream* von *rapD*. Die entsprechende TSS ist in Grafik A mit einem blauen Pfeil markiert. Grafik D zeigt die *upstream* Region vor *phrD*. Die entsprechende TSS ist in Grafik A mit einem grünen Pfeil markiert. In Grafik B, C und D sind mögliche *patterns* für Promotoren rot markiert

Die Expressionsprofile von *rapD* und *phrD* zeigen sehr einheitliche Aktivität in allen Phasen. Das *rapD* Gen scheint über zwei Promotoren zu verfügen, einen SigA Promotor (**TTCACA**-N18-**TATGAT**-N6) und einen SigH Promotor (**AGGCATA-**N15-**GAAT-**N10). Das SigA *pattern* weicht in der Distanz zwischen -10 und -35 Box um eine Base vom beschriebenen Konsensus ab. Außerdem ist im -35 *pattern* die vierte Position kein Guanin.

Das SigH *pattern* hat eine Base mehr Abstand zum Transkriptionsstartpunkt als im Konsensus beschrieben und das -35 *pattern* hat an Position vier kein Adenin. Auffällig ist dass die Basenaktivitäten hinter beiden Promotoren stark abfallen. Dies geht in einigen Fällen so weit dass es für manche Basen keine Abdeckung gab und somit Lücken im Transkript enstehen würden. Das *phrD* Gen hat in seinem *upstream* Bereich einen Kandidaten für einen SigH Promotor (**AGGAATT**-N15-**GAAT**-N9). Dieses *pattern* ist identisch zum Konsensus. Der Promotor zeigt in allen Phasen Aktivität.

Abb. 41 zeigt die Expressionsprofile der *rapK* und *phrK* Gene in den Phasen M1-M5.

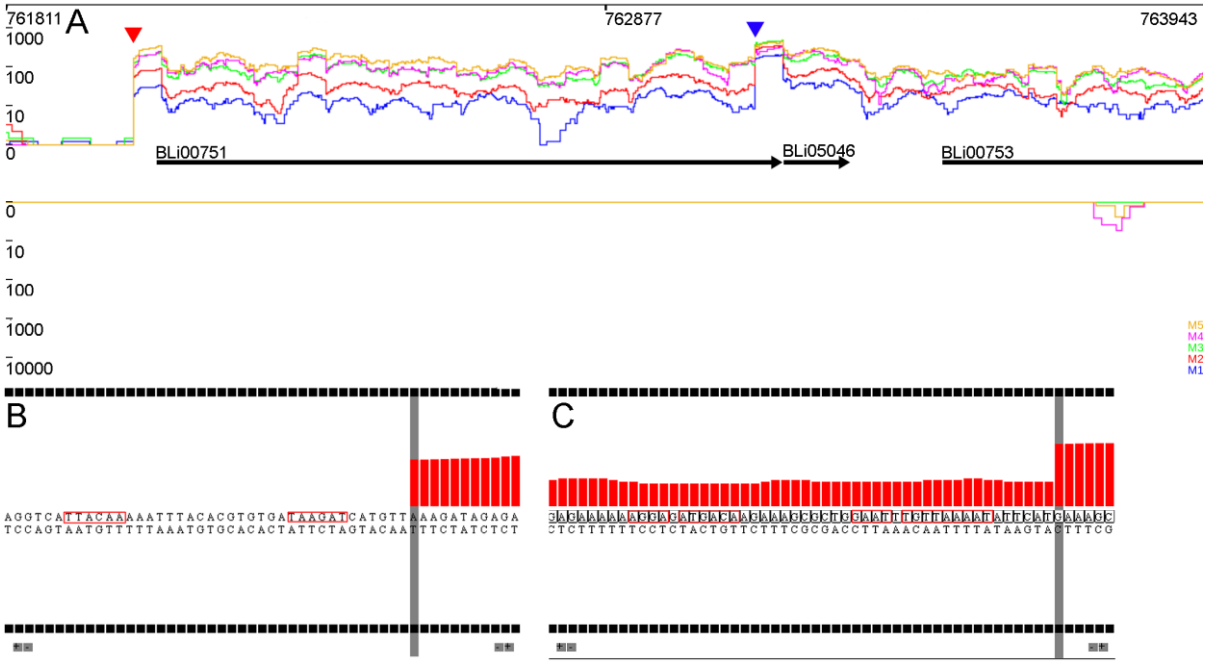

Abb. 41: Expressionsprofile und *upstream* Regionen von *rapK*(BLi00751) und *phrK* (BLi05046)
Grafik A zeigt die Expressionsprofile der Gene. Grafik B zeigt die *upstream* Region vor *rapK*. Die entsprechende TSS ist in Grafik A mit einem roten Pfeil markiert. Grafik C zeigt die *upstream* Region vor *phrK*. Die entsprechende TSS ist in Grafik A mit einem blauen Pfeil markiert. In Grafik B und C sind mögliche *patterns* für Promotoren rot markiert

Anhand der Expressionsprofile lässt sich zeigen, dass *rapK* und *phrK* in allen Phasen aktiv transkribiert werden. Während der Phase M1 zeigt sich die geringste Aktivität. In Phase M2 ist die Aktivität gegenüber M1 erhöht. Während der Phasen M3-M5 zeigt sich die höchste Aktivität wobei diese drei Phasen in etwa gleich starke Transkription für diese Gene aufweisen. Im *upstream* Bereich von *rapK* zeigt sich ein möglicher SigA Promotor (**TTACAA**-N16-**TAAGAT**-N6). Dieser Promotor weicht im -35 *pattern* an Position drei vom Konsensus ab, wo es ein Adenin anstatt eines Guanin aufweist. Vor dem *phrK* Gen liegt ebenfalls ein möglicher SigA Promotor (**ATGACA**-N16-**TG**T**TAAAAT**-N6) sowie ein SigH möglicher Promotor (**AGGA**GA**T**-N15-**GAAT**-N16). Das SigA *pattern* weicht in der -35 Box vom Konsensus ab wo es ein Adenin anstatt eines Thymins hat. Das -10 *pattern* ist um ein TG erweitert. Solche TG-Erweiterungen können schlechte -35 Boxen kompensieren oder

sogar komplett unnötig machen (Kumar A, Malloch RA, Fujita N, Smillie DA, Ishihama A, 1993). Das SigH *pattern* passt zum SigH Konsensus, liegt aber zu weit vom TSS entfernt.

Die NPKM-Werte der *rap* und *phr* Gene sind in Tabelle 26 aufgelistet. Anhand der Werte lässt sich zeigen, dass *B. licheniformis* DSM13 die *rap* und *phr* Gene sehr unterschiedlich verwendet. Die Gene *rapA1* und *rapH* zeigen fast keine Aktivität während *rapK* und *rapI* in über den Verlauf der Fermentation hochgeregelt werden. Im Gegensatz dazu scheinen *rapA2* und *rapG* runterreguliert zu werden. Die Gene *rapD* und *rapI* dagegen zeigen eine relativ stabile Expressionsrate mit Ausnahme der Phase M1 wo *rapI* noch wenig Aktivität zeigt. Die *phr* Gene sind im Allgemeinen viel stärker exprimiert als die entsprechenden *rap* Gene. Außerdem werden sie alle, mit Ausnahme von *phrA1,* über die Zeit hochreguliert. Allein *phrA1* wird in den späteren Phasen weniger stark exprimiert.

Tabelle 26: NPKM-Werte der *rap* und *phr* Gene in den Phasen M1-M5

Gen	M1 NPKM	M2 NPKM	M3 NPKM	M4 NPKM	M5 NPKM
rapA1	7	2	3	8	3
rapA2	172	174	67	50	47
rapG	120	82	29	17	11
rapH	5	3	3	8	2
rapI	32	214	354	307	291
rapD	133	125	90	110	77
rapK	77	190	587	628	854
phrA1	108	185	139	76	55
phrA2	309	782	1043	701	807
phrG	195	343	629	416	458
phrI	63	298	964	1308	1095
phrD	1153	1741	2829	2077	1732
phrK	429	770	1375	1205	1567

In der Tabelle 27 werden die Ratios der *phr* Gene zu den entsprechenden *rap* Genen aufgelistet. Da das *rapH* Gen kein erkennbares *phrH* Gen hat, wird es in dieser Tabelle nicht aufgelistet. Da die *phr* Gene Repressoren für ihre entsprechenden *rap* Gene sind, sollte ein hohes Verhältnis von *phr* Gen zu *rap* Gen darauf hindeuten, dass das *rap* Gen wahrscheinlich reprimiert wird während bei einem kleinen Verhältnis es wahrscheinlicher ist, dass das rap Gen aktiv ist.

Tabelle 27: Ratios der *phr* Gene zu den entsprechenden *rap* Genen

Gen	M1 Ratio	M2 Ratio	M3 Ratio	M4 Ratio	M5 Ratio
phrA1	15,43	92,5	46,33	9,5	18,33
phrA2	1,8	4,49	15,57	14,02	17,17
phrG	1,63	4,18	21,69	24,47	41,64
phrI	1,97	1,39	2,72	4,26	3,76
phrD	8,67	13,93	31,43	18,88	22,49
phrK	5,57	4,05	2,34	1,92	1,83

6.5 *bsrG* Toxin/Anti-toxin System

Das *bsrG*/SR4 ist ein Toxin/Anti-toxin System in Bacillen (Jahn and Brantl, 2013). *BsrG* ist ein kleines Toxin das bei ausreichender Bildung zur Zelllyse führt. Die SR4 RNA bindet an die *bsrG* mRNA und blockiert dabei die Shine-Dalgarno Sequenz für *bsrG*. Zusätzlich wird doppelsträngige RNA schneller durch RNAsen abgebaut. Dadurch übt die SR4 RNA zwei regulatorische Effekte aus nämlich Blockierung der Translation und beschleunigter Abbau der *bsrG* mRNA. Die *bsrG* mRNA ist stabiler als die SR4 RNA, jedoch hat die SR4 einen etwa 10 mal stärkeren Promotor als *bsrG* (Jahn and Brantl, 2013; Jahn *et al.*, 2012) womit ein Titrationseffekt entsteht.

Tabelle 28 zeigt die mittels Rfam-Kovarianzmodell gefundenen *bsrG*/SR4 Kandidaten. Der *trusted-cutoff* des Modells liegt bei 40,1.

Tabelle 28: Vorhersagen für *bsrG*/SR4 Toxin/Anti-Toxin Kandidaten mittels Rfam Kovarianzmodellen

Phase	Name des Kandidaten	Koordinaten im Genom	Score
1	>sRNA_80	652.891-653.178	58,56
2	>sRNA_66	652.890-653.178	58,56
	>UTR3_428	1.298.421+1.300.414	61,16
3	>sRNA_69	652.890-653.177	58,56
4	>sRNA_96	652.889-653.178	58,56
	>UTR3_457	1.298.421+1.300.549	148,24
	>UTR3_1275	3.484.971+3.485.394	43,47
5	>sRNA_70	652.889-653.178	58,56
	>UTR3_461	1.298.421+1.300.441	120,20

Anhand dieser Vorhersagen konnten drei *bsrG*/SR4 Kandidaten identifiziert werden, aufgelistet in Tabelle 29.

Tabelle 29: Koordinaten der *bsrG* und SR4 Kandidaten

Name	**Koordinaten des putativen SR4**	**Koordinaten des putativen *bsrG***
bsrG/SR4_1	652.850+653.022	652.890-653.178
bsrG/SR4_2	1.300.088+1.300.313	1.300.192-1.300.453
bsrG/SR4_3	3.485.163+ 3.485.310	3.485.194-3.485.453

Mit den Koordinaten lassen sich die Längen der Transkripte berechnen und mit den Werten von Jahn und Brantl vergleichen. Jahn und Brantl sagen für *B. subtilis* Längen von 294 Basen für die *bsrG* mRNA und 180 Basen für die *sr4* sRNA voraus. Tabelle 30 gibt die längen der Transkripte für die *B. licheniformis* DSM13 *bsrG*/SR4 Kandidaten an.

Tabelle 30: Länge der *bsrG*/SR4 Transkripte in *B. licheniformis* DSM13

Name	**Länge *sr4* sRNA**	**Länge *bsrG* mRNA**
bsrG/SR4_1	173	283
bsrG/SR4_2	225	258
bsrG/SR4_3	156	263

Im Allgemeinen weichen die Längen in *B. licheniformis* DSM13 geringfügig von denen in *B. subtilis* ab, wobei sie bis auf einen Fall kürzer sind.

Abb. 42 zeigt die genomischen Kontexte der vorhergesagten *bsrG*/SR4 Kandidaten.

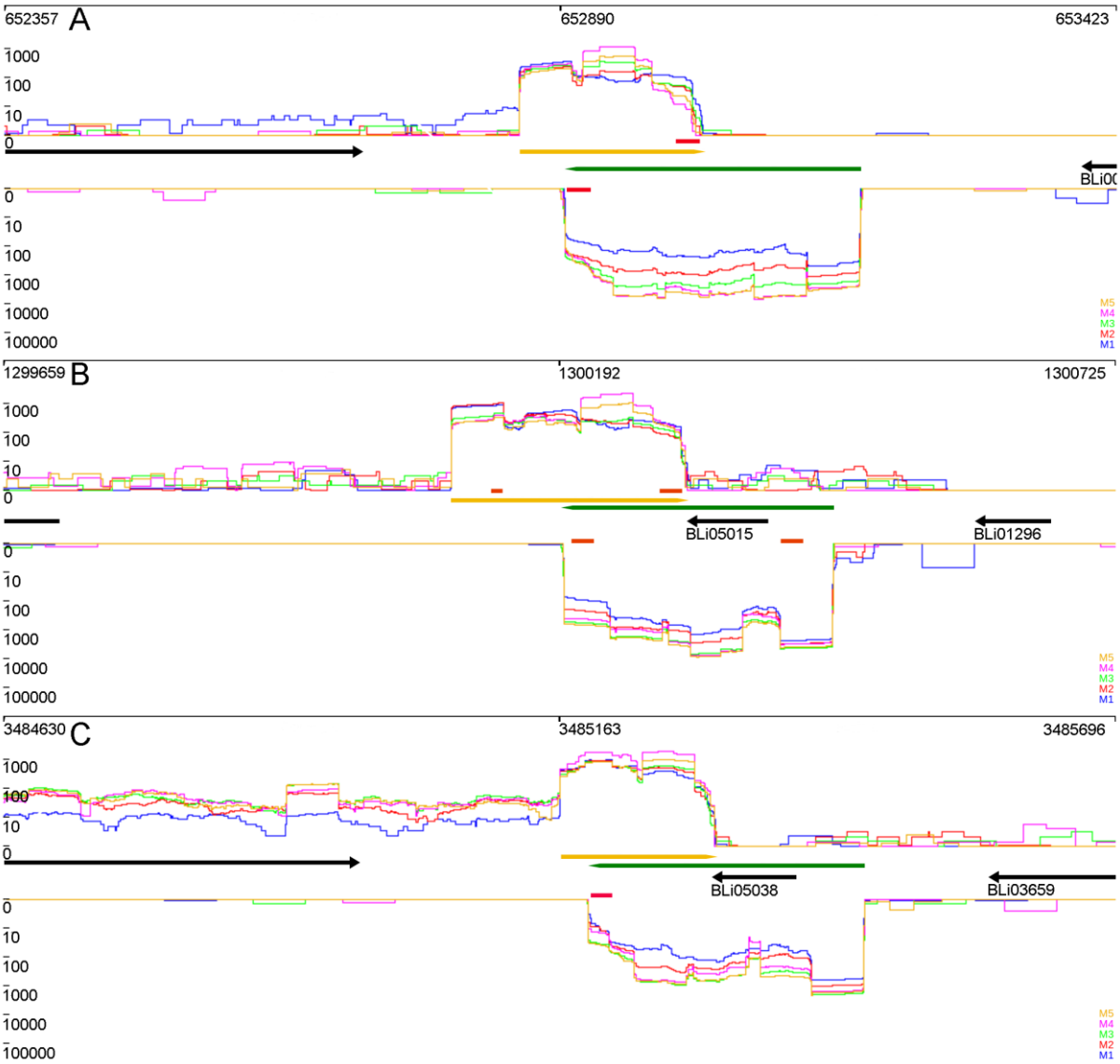

Abb. 42: Transkriptionale Aktivitäten von vorhergesagten *bsrG*/SR4 Kandidaten in den Phasen M1-M5 im genomischen Kontext
BsrG Kandidaten sind mit grünen Pfeilen und SR4 Kandidaten mit gelben Pfeilen markiert. Mit TransTermHP vorhergesagte Terminatoren sind mit roten Kästchen markiert. Grafik A zeigt den genomischen Kontext von *bsrG*/SR4_1, Grafik B zeigt den genomischen Kontext von *bsrG*/SR4_2, Grafik C zeigt den genomischen Kontext von *bsrG*/SR4_3

Anhand der Expressionsprofile der *bsrG*/SR4 Kandidaten lässt sich sagen, dass alle Kandidaten in allen Phasen Aktivität zeigen. Die *bsrG* Kandidaten zeigen ähnliche Expressionsstärke wie die SR4 Kandidaten wobei die Unterschiede zwischen den Phasen bei SR4 etwas größer sind.

Tabelle 31 zeigt die NPKM-Werte der *bsrG*/SR4 Kandidaten und die Verhältnisse von *bsrG* gegenüber den SR4 Transkripten.

Tabelle 31 NPKM-Werte und Verhältnisse der NPKM-Werte von *bsrG*/SR4 Kandidaten

Gen	**M1 NPKMs**	**M2 NPKMs**	**M3 NPKMs**	**M4 NPKMs**	**M5 NPKMs**
bsrG1	5277	8385	16630	15035	18873
SR4_1	2412	2351	1499	4060	2214
bsrG2	1076	1949	4564	3419	4324
SR4_2	2785	2922	2997	7593	3791
bsrG3	1245	4190	11366	22757	23746
SR4_3	940	778	1141	2543	1350
***bsrG*/SR4 pair**	**M1 Ratio**	**M2 Ratio**	**M3 Ratio**	**M4 Ratio**	**M5 Ratio**
bsrG1/SR4_1	2,2	3,6	11,1	3,7	8,5
bsrG2/SR4_2	0,4	0,7	1,5	0,5	1,1
bsrG3/SR4_3	1,3	5,4	10,0	8,9	17,6
Gen	**R1 NPKMs**	**R2 NPKMs**	**R3 NPKMs**	**R4 NPKMs**	**R5 NPKMs**
bsrG1	9189	11277	20506	11488	17553
SR4_1	1169	808	874	3327	2166
bsrG2	1374	2804	4555	2939	3532
SR4_2	1230	1593	3215	5841	4060
bsrG3	2848	6300	12390	27162	28914
SR4_3	669	271	458	1792	1404
***bsrG*/SR4 pair**	**R1 Ratio**	**R2 Ratio**	**R3 Ratio**	**R4 Ratio**	**R5 Ratio**
bsrG1/SR4_1	7,9	14,0	23,5	3,5	8,1
bsrG2/SR4_2	1,1	1,8	1,4	0,5	0,9
bsrG3/SR4_3	4,3	23,2	27,1	15,2	20,6
Gen	**L1 NPKMs**	**L2 NPKMs**	**L3 NPKMs**	**L4 NPKMs**	**L5 NPKMs**
bsrG1	12873	19090	20568	16515	19117
SR4_1	1047	719	1765	2832	2038
bsrG2	2667	4994	5020	3064	3880
SR4_2	979	1394	5029	7175	4240
bsrG3	6861	10252	13983	31303	29669
SR4_3	500	422	1524	1832	1192
***bsrG*/SR4 pair**	**L1 Ratio**	**L2 Ratio**	**L3 Ratio**	**L4 Ratio**	**L5 Ratio**
bsrG1/SR4_1	12,3	26,6	11,7	5,8	9,4
bsrG2/SR4_2	2,7	3,6	1,0	0,4	0,9
bsrG3/SR4_3	13,7	24,3	9,2	17,1	24,9

Anhand der Werte in Tabelle 31 zeigt sich, dass die Replikate voneinander abweichen. Tendenziell werden die *bsrG* Gene um ein vielfaches stärker exprimiert werden als die entsprechenden SR4 Kandidaten, wobei das *bsrG*/SR4_2 Paar einige Ausnahmen in den Phasen 4 und 5 sowie M1 und M2 aufweist. Dies wiederspricht den Verhältnissen die in *B. subtilis* für *bsrG* und SR4 beschrieben wurden (Jahn *et al.*, 2012), wo für SR4 ein etwa 6-10 mal stärkerer Promotor als für *bsrG* gefunden wurde.

6.6 Phasenabhängige Expressionsprofile

Anhand des Expressionsverhaltens von Genen über den Verlauf der Fermentation lassen sich bedingungs- und prozessspezifische Muster erkennen. Dies wird an drei Beispielen für differentielle Expression deutlich. Das erste Beispiel ist das *hag* Gen welches für eine Strukturkomponente der Flagellen kodiert. Die Transkription dieses Gens wird durch den SigD σ-Faktor, einen σ-Faktor für Flagellengene (siehe 2.1.1), kontrolliert (Mirel and Chamberlin, 1989). Abb. 43 zeigt die Expressionsprofile des *hag* Gens während der Phasen M1 bis M5 sowie den *upstream* Bereich des *hag* Transkripts.

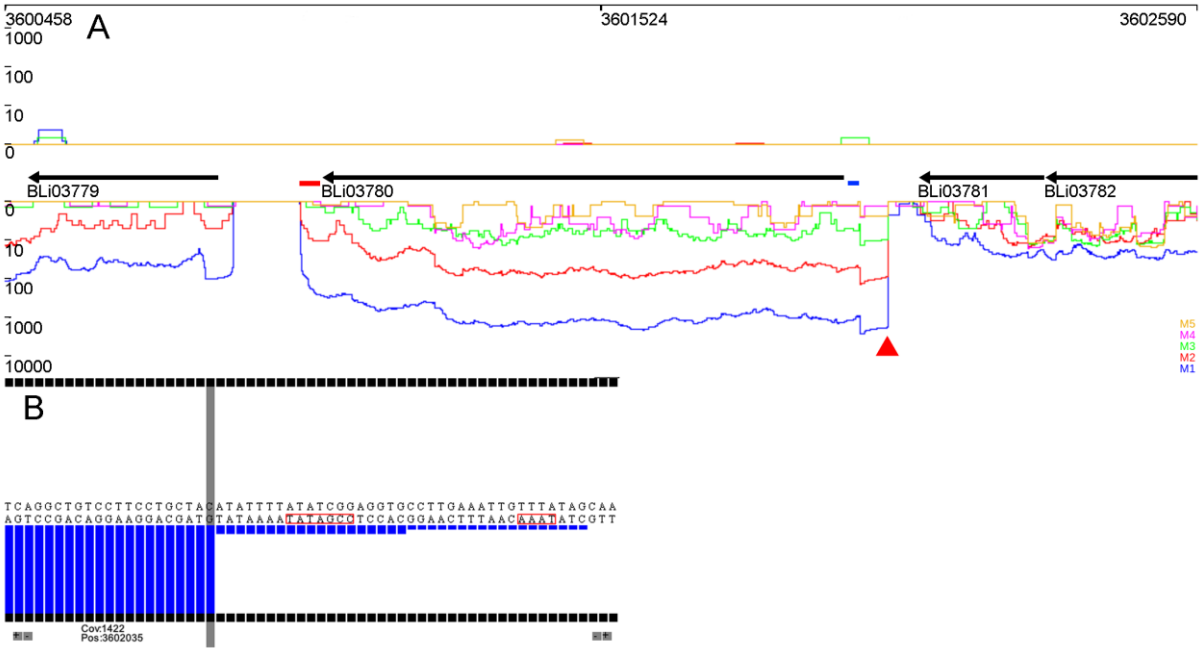

Abb. 43: Transkriptionale Aktivität des *hag* Gens (BLi03780) in den Phasen M1 bis M5 im genomischen Kontext
Grafik A zeigt die Expressionsprofile des *hag* Gens. Das rote Kästchen markiert den mit TransTermHP vorhergesagte Terminator hinter *hag*, das blaue Kästchen markiert die manuell vorhergesagte Shine-Dalgarno Sequenz für *hag*. Grafik B zeigt den *upstream* Bereich vom vorhergesagten TSS von *hag.* Die grau markierte Base in Grafik B ist in Grafik A mit einem roten Pfeil markiert. Innerhalb von Grafik B sind putative Promotoren rot markiert

In Abb. 43 (B) sind die -10 und -35 *patterns* für SigD gut erkennbar (**TAAA**-N16-**CCGATAT**-N7). Dies entspricht dem Konsensus des SigD Promotors (**TAAA**-N14/16-g**CCGATAT**) (Sonenshein *et al.*, 2002). In der Phase M1 zeigt sich die größte Aktivität des *hag* Gens und

nimmt im Verlauf bis M5 kontinuierlich ab, wobei M3 bis M5 sehr nahe beieinander liegen. Dies zeigt sich ebenfalls anhand der NPKM Werte, aufgelistet in Tabelle 32.

Tabelle 32: NPKM-Werte des *hag* Gens in den Phasen 1 bis 5

Replikat	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
M	6635	325	37	23	9
R	488	79	6	8	10
L	106	51	3	13	13

Die Replikate bestätigen dieses Expressionsverhalten, unterscheiden sich jedoch in den Verhältnissen der Phasen untereinander. Das *hag* Gen wird also im Verlauf der Fermentation herunterreguliert. Anhand der NPKM-Werte in Tabelle 33 des *sigD* Gens wird deutlich, dass *B. licheniformis* unter den Fermentationsbedingungen anscheinend keine Flagellen ausbildet.

Tabelle 33: NPKM Werte des *sigD* Gens in den Phasen M1 bis M5

Gen	NPKM M1	NPKM M2	NPKM M3	NPKM M4	NPKM M5
sigD	160	11	6	9	5

Das zweite Beispiel ist das *spoIVA* Gen, dessen Produkt bei der Bildung des Sporenmantels während der Sporulation beteiligt ist (McKenney *et al.*, 2013). Die Expression wird durch SigE kontrolliert (Eichenberger *et al.*, 2003). Der Konsensus für SigE ist demnach (**T**C**AT**ATT-N15-**CATA**CGA**T**-N6). Abb. 44 zeigt das Expressionsprofil von *spoIVA* in den Phasen M1 bis M5 sowie die *upstream*-Region vom putativen TSS vor *spoIVA*.

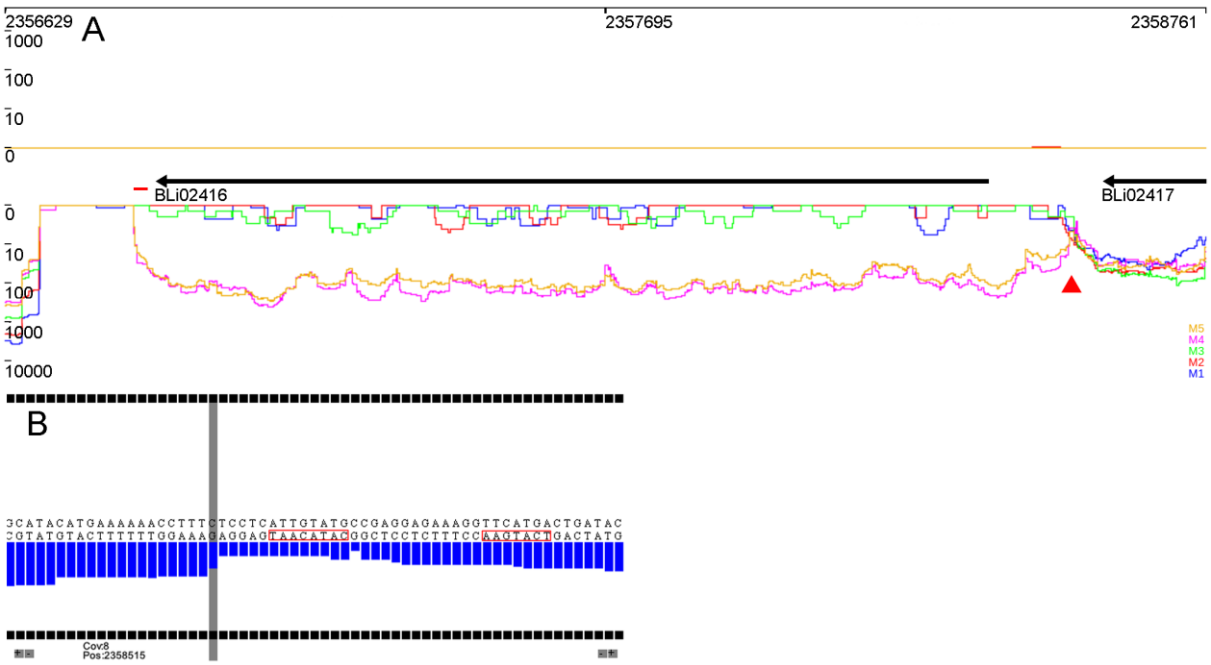

Abb. 44: Transkriptionelle Aktivität von *spoIVA* (BLi02416) im genomischen Kontext
Grafik A zeigt die Expressionsprofile von *spoIVA* in den Phasen M1 bis M5. Das rote Kästchen markiert einenmit TransTermHP vorhergesagten Terminator. Grafik B zeigt die upstream Region von *spoIVA* wobei die grau markierte Position in Grafik A mit einem roten Pfeil markiert ist. Putative Promotor *patterns* sind rot umrandet

In Abb. 44 (B) sind SigE ähnliche -10 und -35 *patterns* zu erkennen (**TACTGAA**-N13-**CATACAAT**-N5). Während die Sequenzen gut zum Konsensus passen, sind die Abstände zwischen -35 und -10 *pattern* sowie zwischen TSS und -10 *pattern* etwas zu kurz. Während der Phasen M1 bis M3 zeigt sich in den Expressionsprofilen fast keine Aktivität. Erst in den Phasen M4 und M5 zeigt sich transkriptionelle Aktivität. Dabei scheint M4 die höhere Aktivität als M5 zu zeigen. Tabelle 34 zeigt die NPKM Werte in den Phasen 1 bis 5 der drei Replikate.

Tabelle 34: NPKM Werte des *spoIVA* Gens in den Phasen 1 bis 5

Replikat	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
M	6	6	9	915	668
R	8	5	16	1011	181
L	3	10	89	1394	779

Die Replikate bestätigen das Expressionsverhalten wobei auch hier die Verhältnisse zwischen den Replikaten schwanken. Das *spoIVA* Gen zeigt nur während der stationären Phasen transkriptionelle Aktivität.

Das dritte Beispiel ist das Inositol Operon, welches zehn Gene umfasst. Die Gene dieses Operons werden zur Verarbeitung von myo-Inositol, einem zyklischen 6fach Zuckeralkohol

benötigt. Inositol kommt im Boden vor und kann von Mikroorganismen als C-Quelle verwendet werden (Yoshida *et al.*, 1997). Abb. 45 zeigt das Expressionsprofil des Inositol Operons in den Phasen M1 bis M5 sowie die *upstream* Region vor dem putativen TSS des Operons.

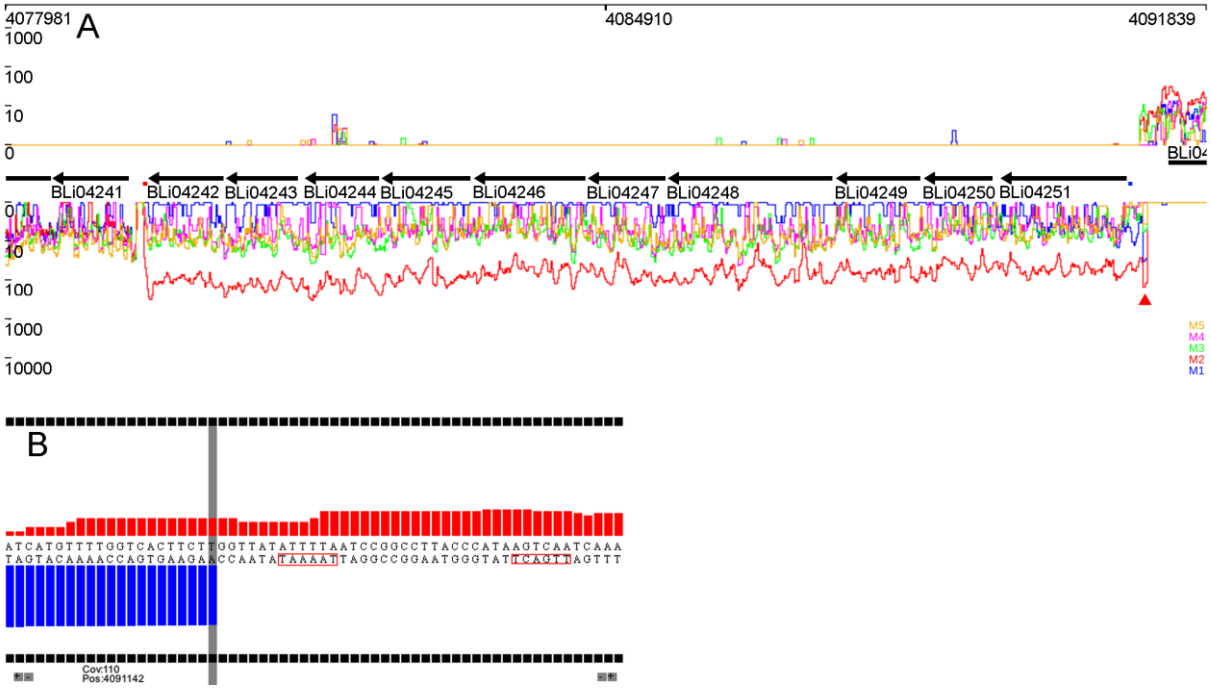

Abb. 45: Transkriptionelle Aktivitäten Inositol Operons (BLi04242 bis BLi04251) in den Phasen M1 bis M5 im genomischen Kontext
Grafik A zeigt die Expressionsprofile in den Phasen M1 bis M5. Das rote Kästchen markiert einen mit TransTermHP vorhergesagten Terminator, das blaue Kästchen markiert eine manuell vorhergesagte Shine-Dalgarno Sequenz. Grafik B zeigt die upstream Region vor der putativen TSS des Inositol Operons. Die grau markierte Base ist in Grafik A mit einem roten Pfeil markiert

In Abb. 45(B) sind -10 und -35 *patterns* für SigA zu erkennen (**TTGACT**-N17-**TAAAAT**-N6) welche gut zum Konsensus eines SigA Promotors passen. Das Operon zeigt während M1 die geringste Aktivität und während Phase M2 die höchste. In Phase M3 nimmt die Aktivität gegenüber M2 wieder ab und fällt in M4 und M5 geringfügig weiter ab. Tabelle 35 zeigt die NPKM Werte der Inositol Operon Gene in den Phasen 1 bis 5. Die Replikate unterscheiden sich in ihren Aussagen geringfügig.

Tabelle 35: NPKM Werte der Inositol Operon Gene in den Phasen 1 bis 5 der drei Replikate M,R,L

Gen	Phase M1	Phase M2	Phase M3	Phase M4	Phase M5
BLi04242	9	740	57	38	61
BLi04243	10	874	79	50	57
BLi04244	6	920	91	41	67
BLi04245	3	569	45	36	34
BLi04246	4	392	40	29	30
BLi04247	5	489	66	46	52
BLi04248	7	484	71	49	61
BLi04249	6	407	52	39	41
BLi04250	7	320	38	30	32
BLi04251	25	347	40	27	33
Gen	**Phase R1**	**Phase R2**	**Phase R3**	**Phase R4**	**Phase R5**
BLi04242	83	85	66	49	46
BLi04243	158	100	85	50	53
BLi04244	126	105	89	63	47
BLi04245	122	60	50	32	37
BLi04246	127	46	34	20	24
BLi04247	257	92	74	42	49
BLi04248	259	77	74	42	44
BLi04249	259	73	54	29	37
BLi04250	161	63	51	36	36
BLi04251	78	56	42	22	22
Gen	**Phase L1**	**Phase L2**	**Phase L3**	**Phase L4**	**Phase L5**
BLi04242	395	59	117	40	54
BLi04243	445	76	148	58	60
BLi04244	410	61	126	56	68
BLi04245	228	43	101	41	40
BLi04246	150	32	62	25	30
BLi04247	243	62	124	43	57
BLi04248	235	56	111	41	57
BLi04249	196	45	90	31	45
BLi04250	158	49	97	37	42
BLi04251	82	37	70	20	31

Die Replikate zeigen die höchste Aktivität des Operons in den Phase 1 bis 2 wobei das M Replikat die höchste Aktivität in Phase 2 zeigt und das L und R Replikat die höchste Aktivität in Phase 1 zeigen. Das R Replikat zeigt zwischen Phase 1 und 2 nur geringfügige Differenzen im Gegensatz zu den L und M Replikaten wo die Differenzen ausgeprägter ist. Somit lässt sich zeigen, dass das Inositol Operon in den frühen Phasen 1 und 2 aktiv ist und in seiner Aktivität über den Verlauf der Fermentation wieder abnimmt.

7 Promotorvorhersage

Durch die Möglichkeit, mittels TraV die Startpunkte der Transkription (TSS) vorherzusagen, lässt sich auf die Positionen der Promotoren schließen, welche diese TSS bedingt haben. Zu diesem Zweck wurde ein Programm namens „Nimmersatt“ entwickelt, dass aufbauend auf den TraV TSS Vorhersagen die entsprechenden Sequenzen *upstream* vom TSS sammelt und diese mittels MEME nach konservierten *patterns* durchsucht.

Implementiert wurde dieses Programm in Java. Ausgehend von den TSS Kandidaten extrahiert Nimmersatt die *upstream*-Sequenzen bis 50 Basen vom TSS. Diese werden dann mittels MEME (Bailey *et al.*, 2006) nach *patterns* durchsucht. Eine Einschränkung von MEME verlangt hierbei ein spezielles Vorgehen. MEME verlangt eine vorgegebene Menge an zu erwartenden Motiven. Da aber nicht klar ist, wie viele unterschiedliche *patterns* zu erwarten sind, wird MEME dazu verwendet, nur ein gut konserviertes *pattern* zu suchen. Die von MEME zugewiesenen Sequenzen werden dann aus dem *pool* an verfügbaren TSS Sequenzen entfernt und eine neue Suche wird gestartet. Dies geschieht solange bis entweder alle Sequenzen aus dem TSS Kandidatenpool einem *pattern* zugeordnet worden konnten oder aber MEME nicht mehr in der Lage ist, verbliebene Kandidaten einem neuen *pattern* zuzuordnen. Abb. 46 zeigt ein Flussdiagramm für den Nimmersatt Algorithmus.

Für die erhaltenen *patterns* generiert MEME WebLogos (Crooks *et al.*, 2004). Zusätzlich verwendet Nimmersatt die Annotationen des zugehörigen Genoms um jeweils das erste Gen *downstream* vom TSS zu ermitteln. Ein Gen wird einem TSS zugeordnet, wenn es zwischen TSS und dem Start vom Gen keine Unterbrechung der Basenaktivitäten (im Sinne einer Basenaktivität von Null) gibt. Diese Gene werden dann mittels COG (Tatusov *et al.*, 2001) einer COG Kategorie zugeordnet, sodass man eine Abschätzung darüber machen kann, ob gefundene *patterns* spezifische Verteilungen der COG Kategorien aufweisen.

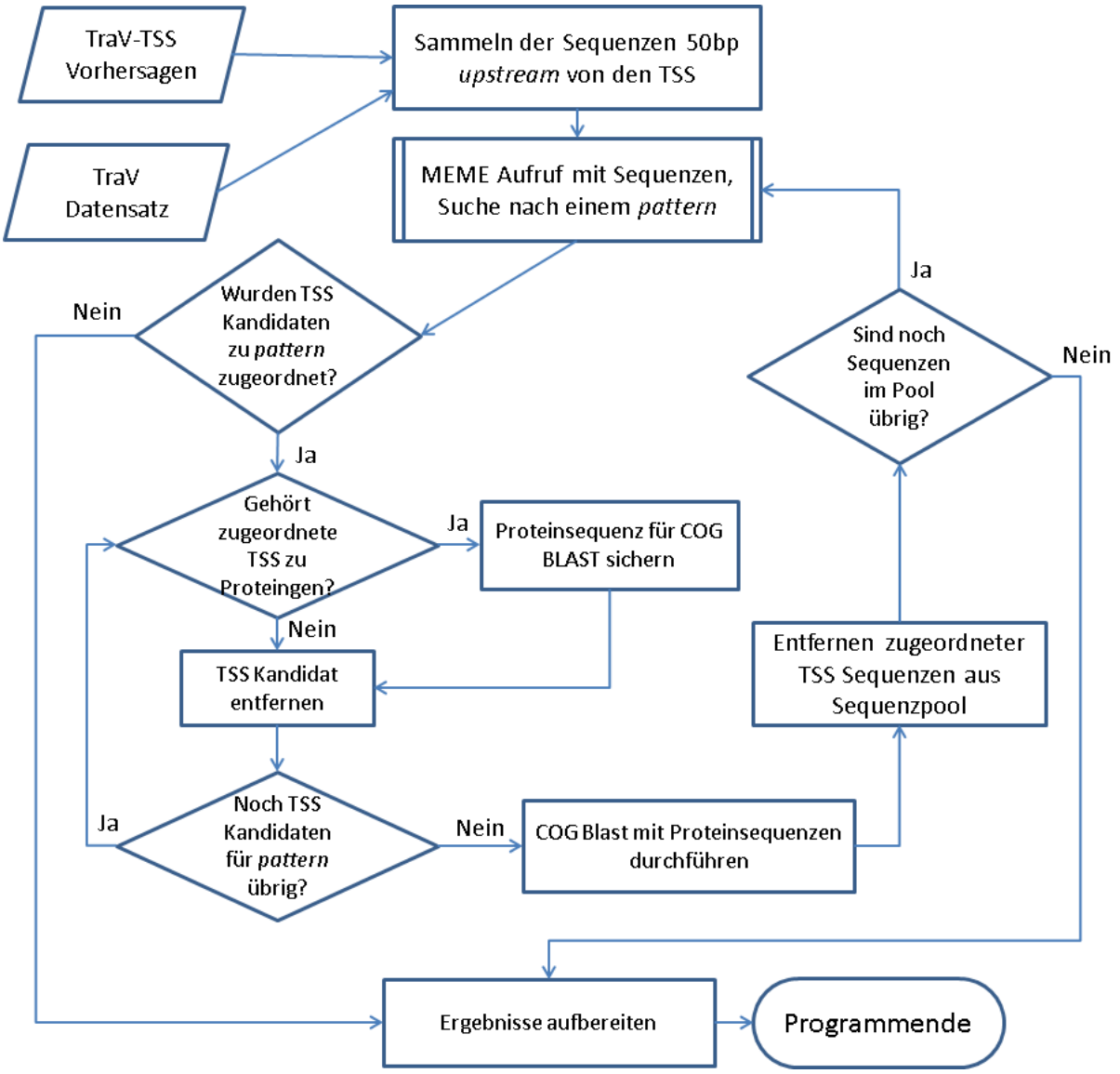

Abb. 46: Flussdiagramm des Nimmersatt Algorithmus

Insgesamt konnten mit dem kompletten TSS Kandidatensatz 221 *patterns* identifiziert werden. Von den 221 *patterns* wurden manuell jene selektiert, die für σ-Faktor Bindestellen, also -35 und -10 *patterns*, typisch sind. Das *pattern* muss demnach zwei konservierte *loci* mit einem Abstand von etwa 15 bis 18 Basen zueinander aufweisen in dem möglichst keine konservierten Basen vorkommen. Abb. 47 zeigt die gefundenen *patterns* aus dem vollen Datensatz welche zu diesem Schema passen.

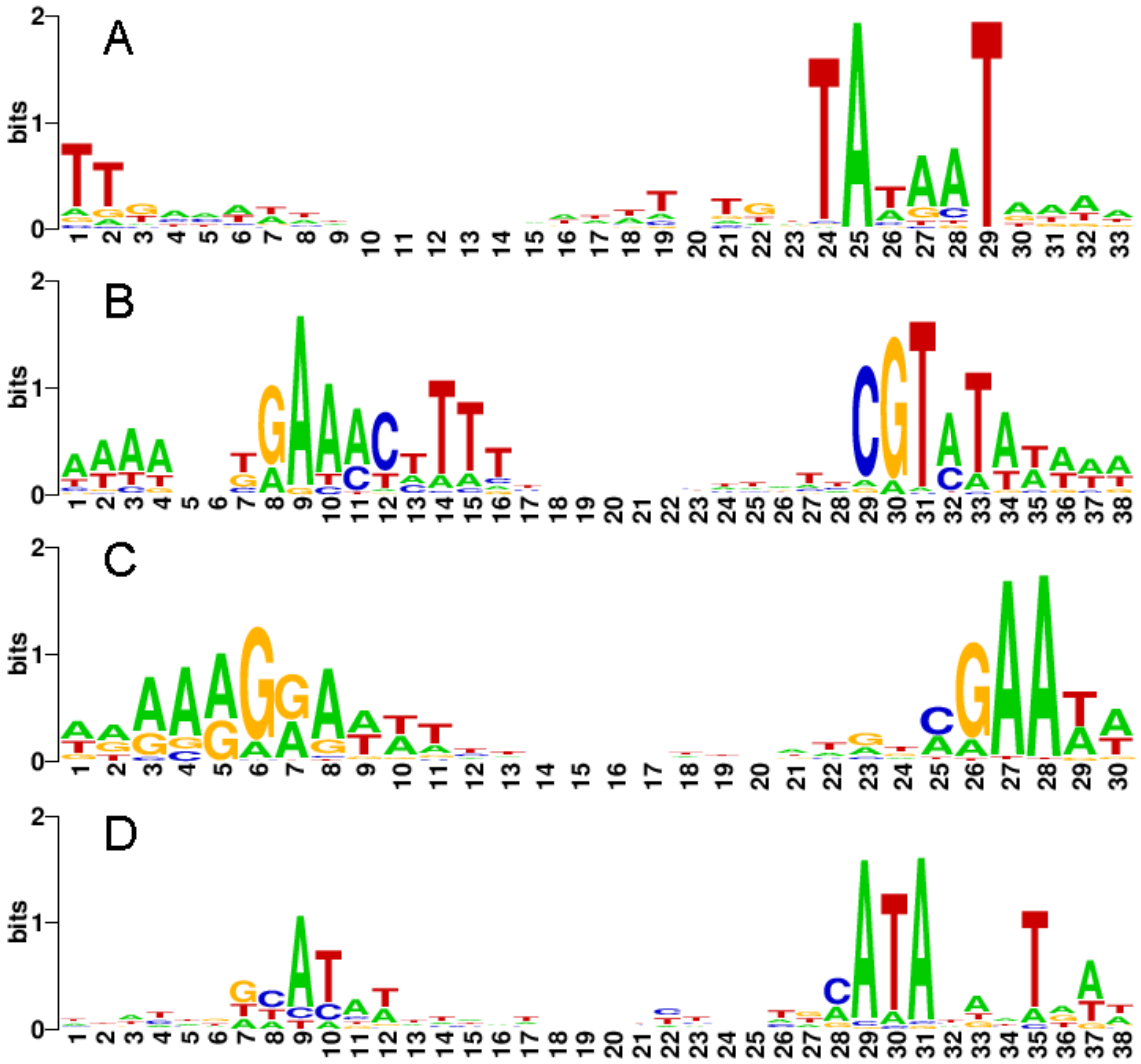

Abb. 47: Mit Nimmersatt gefundene Promotor *patterns* im gesamten TSS Datensatz
Grafik A zeigt das *pattern* eines SigA Promotors. Grafik B ist eine Kombination aus verschiedenen ECF-σ-Faktor *patterns*. Grafik C zeigt das *pattern* eines SigH Promotors. Grafik D zeigt eine mögliche Mischung von SigE und SigK *patterns*

Die in Abb. 47 gezeigten *patterns* passen zu mehreren σ-Faktor Promotoren. Das *pattern* in Abb. 47 (A) passt zu einem SigA Promotor wobei das -35 *pattern* schwächer konserviert zu sein scheint als das -10 *pattern*. Beim -10 pattern ist die spezifische Konservierung der -12,-11 und -7 Position (Positionen 24, 25 und 29 in der Grafik) gut erkennbar, wie sie von Feklistov und Darst beschrieben wird (Feklistov and Darst, 2011). Zusätzlich ist die TG Erweiterung des -10 *patterns* erkennbar. Die Abb. 47 (B) zeigt wahrscheinlich kondensierte *patterns* von mehreren ECF-σ-Faktoren (siehe 2.1.3). Mögliche ECF-σ-Faktor *patterns,* die zum vorhergesagten *pattern* passen sind ECF der Gruppe 1, 2, 11, 12, 15, 17, 30 und 31. Die ECF-σ-Faktoren sind in ihren *patterns* sehr ähnlich und MEME ist anscheinend nicht in der Lage, diese *patterns* zu trennen. Diese Ähnlichkeit der *patterns* wurde bereits von Mascher *et al.* beschrieben (Mascher *et al.*, 2007). Die Abb. 47 (C) zeigt ein *pattern*, das zum Konsensus von SigH passt. Die Abb. 47 (D) zeigt ein *pattern*, das wahrscheinlich zwei

σ-Faktoren beinhaltet, nämlich SigE und SigK, welche sehr ähnliche Promotoren benutzen (Eichenberger *et al.*, 2003) und (Silvaggi *et al.*, 2006).

Abb. 48. zeigt ein Diagramm, dass die Verteilung der ersten Proteine *downstream* von den vorhergesagten TSS für die SigA Promotoren in COG Kategorien darstellt.

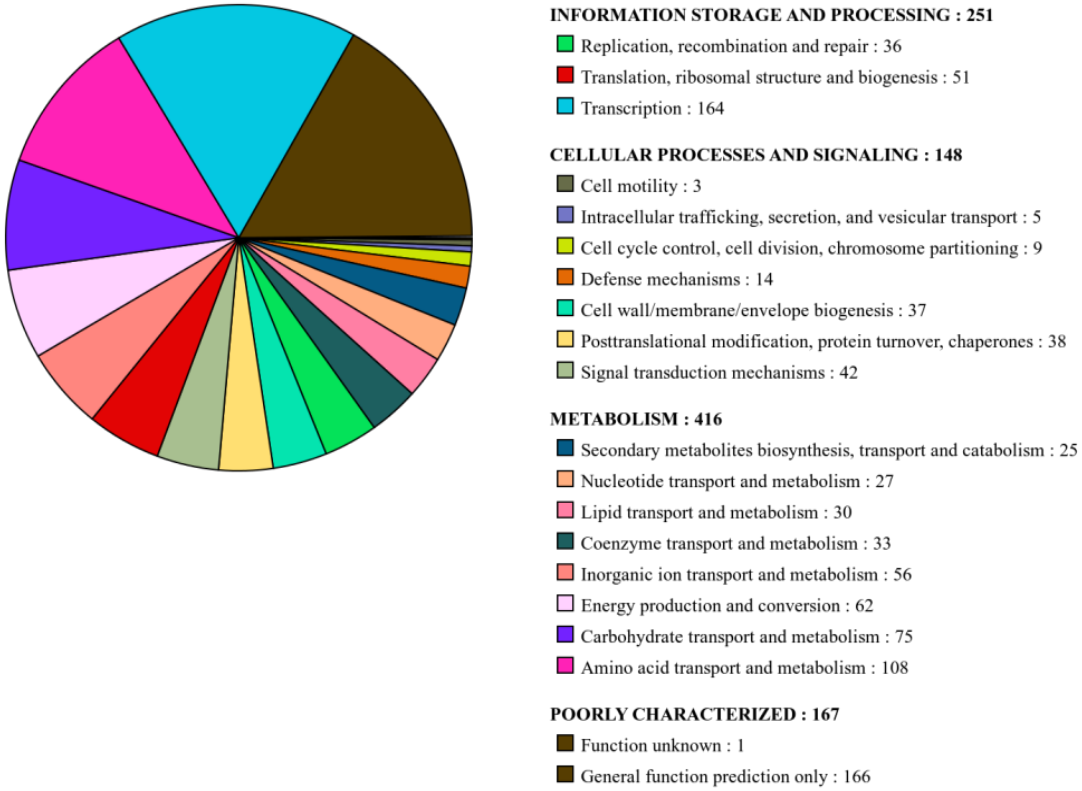

Abb. 48: Übersicht der COG Kategorien der ersten Proteine *downstream* von vorhergesagten TSS für SigA Promotoren
Anhand der Anzahl an Proteinen in den einzelnen COG Kategorien lässt sich sagen, dass der SigA Promotor viele Proteine des Metabolismus und der Informationsverarbeitung, hier vor allem Transkriptions- und Translationsregulatoren, kontrolliert

Für SigA Promotoren wurden insgesamt 1317 Kandidaten gefunden. Für 1136 dieser Kandidaten konnten *downstream* proteinkodierende Gene gefunden werden von denen 982 mittels COG einer Kategorie zugeordnet werden konnten. Nach der Verteilung der COG Kategorien liegt die Mehrzahl der gefundenen Gene in der Kategorie Metabolismus, wobei die meisten Gene hier Funktionen im Aminosäurestoffwechsel, Kohlenstoffstoffwechsel sowie im allgemeinen Energiehaushalt der Zelle erfüllen. Neben dem Metabolismus scheint es viele Gene in der Kategorie Informationsverarbeitung zu geben wobei vor allem Funktionen zur Steuerung der Transkription betroffen sind.

Abb. 49 zeigt ein Diagramm für Proteine *downstream* von TSS Vorhersagen für die ECF-σ-Faktoren.

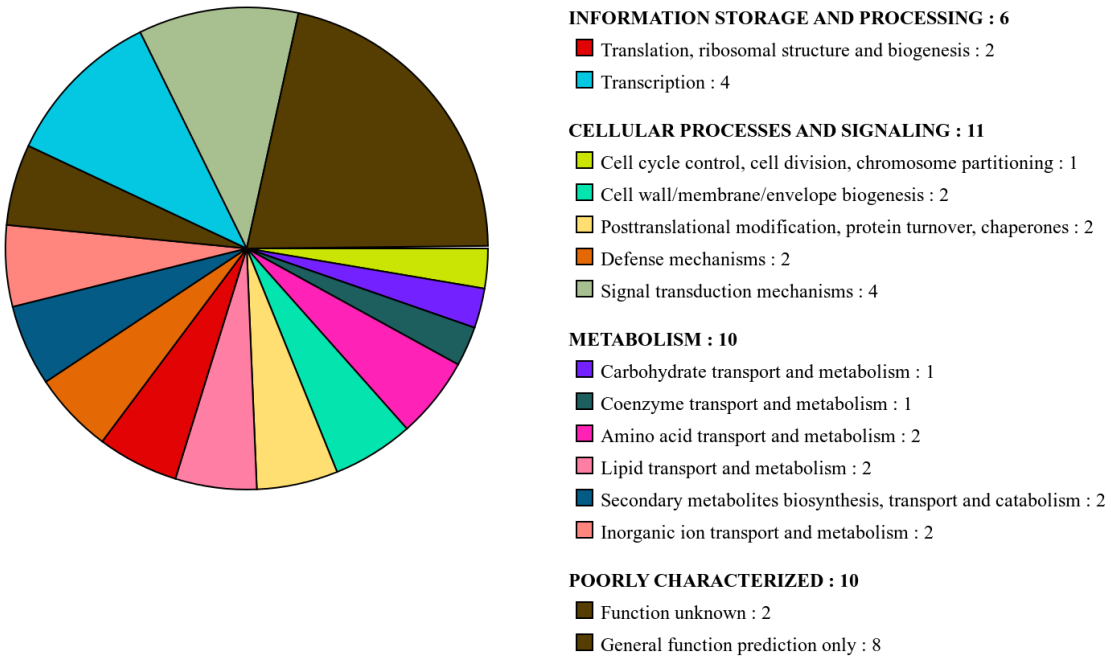

Abb. 49: Übersicht der COG Kategorien der ersten Proteine *downstream* von vorhergesagten TSS für ECF-σ-Faktor Promotoren
Die Proteine, die durch diese σ-Faktoren kontrolliert werden verteilen sich sehr grleichmäßig auf verschiedene COG Kategorien

Für die ECF-σ-Faktor Promotoren wurden insgesamt 61 Kandidaten vorhergesagt von denen 53 zu einem Protein zugeordnet werden konnten. Von diesen 53 Proteinen konnten 37 in COG Kategorien eingeteilt werden. Die Proteine verteilen sich gleichmäßig auf die Kategorien Informationsfluss, Intrazelluläre Prozesse und Metabolismus wobei die am stärksten vertretene Kategorie Proteine für Signal Transduktion und Transkriptionssteuerung sind. Das passt sehr gut zu der erwarteten Klasse von Proteinen, nämlich ECF-σ-Faktoren und deren Regulatoren welche oft autoinduzierend sind. ECFs sind größtenteils an der Zellantwort auf Zellhüll-, Antibiotika- und physikalischen Stress beteiligt (Staroń *et al.*, 2009) Ein Beispiel aus den Vorhersagen ist das *sigW* Gen, welches ein bekannter ECF-σ-Faktor ist. SigW kontrolliert Gene, die an der Zellantwort auf alkalinen Zellhüllstress sowie Zellwandsynthese hemmende Antibiotika beteiligt sind (Cao *et al.*, 2002). Einige dieser Funktionen deuten sich in den COG Kategorien für die ECF-σ-Faktor kontrollierten Gene ab. Abb. 50 zeigt die entsprechenden *upstream* Bereiche vor den TSS dieses Genes im Detail. Das *sigW* Gen verfügt anscheinend über zwei Promotoren, einmal den eigenen SigW Promotor und einen schwachen SigA Promotor, welcher nur während der Phase M1 (Wiegand *et al.*, 2013) Aktivität zu zeigen scheint und in den Replikaten still ist.

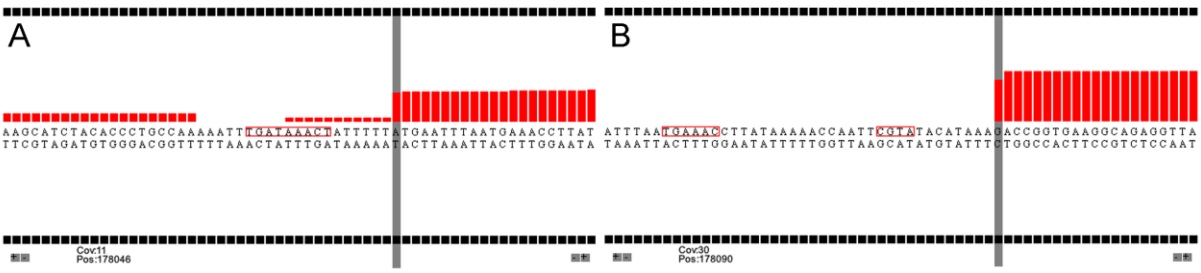

Abb. 50: *Upstream* Regionen von putativen TSS vor *sigW*
Grafik A zeigt den Bereich eines putativen SigA Promotors und Grafik B zeigt einen putativen SigW Promotor.Mögliche Promotor *patterns* sind rot markiert

Der putative SigA Promotor verfügt über ein erweitertes -10 *pattern* (**TG**A**TA**AAC**T**-N6) und kein konserviertes -35 *pattern.* Der putative SigW Promotor dagegen entspricht dem Konsensus (**TGAAAC**-N16-**CGTA**-N8). Solche multiplen Promotoren vor einem Gen geben einen Ansatz für die Analyse von multilayer Regulation.

Abb. 51 zeigt ein Diagramm für Proteine *downstream* von TSS Vorhersagen für SigH Promotoren.

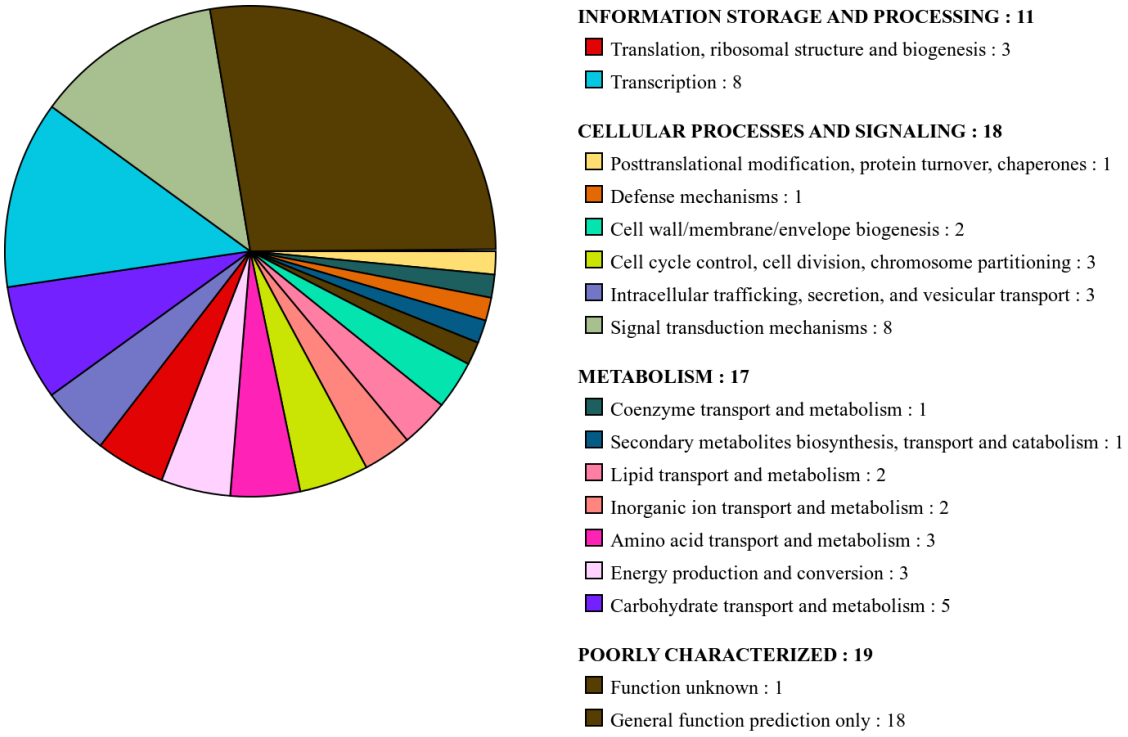

Abb. 51: Übersicht der COG Kategorien der ersten Proteine *downstream* von vorhergesagten TSS für SigH Promotoren
Die meisten, klassifizierbaren Proteine fallen in die COG Kategorien der Signaltransduktion, Transkriptionskontrolle und Kohlenstoffmetabolismus wobei ca. ein viertel der Proteine nicht genauer durch COG klassifiziert werden können

Für die SigH Promotoren wurden insgesamt 104 Kandidaten vorhergesagt von denen 89 zu einem Protein zugeordnet werden konnten. Für 65 dieser Proteine konnten Zuordnungen zu COG Kategorien gemacht werden. Die meisten Proteine konnten den COG Kategorien intrazelluläre Prozesse und Metabolismus zugeordnet werden, wobei etwa ein Viertel nicht genauer charakterisiert werden konnte. Die meisten klassifizierbaren Proteine sind in der Signaltransduktion, der Transkription und dem Kohlenstoffmetabolismus beteiligt. SigH ist beteiligt an der Expression von Genen die an der Einleitung der Sporulation beteiligt sind (Predich *et al.*, 1992).

Basierend auf der Annotation der *downstream* liegenden Proteine sind etwa ein Sechstel dieser Proteine einer Funktion zuzuordnen, welche an der Sporulation beteiligt sein können und somit in das Regulon von SigH passen. In den mit MEME gefundenen Proteingenen mit SigH Promotoren konnten *spoVG*, *citG*, *spoIIA, ftsA, spo0A* und *spo0F* bestätigt werden welche von Predich *et al.* als SigH kontrolliert beschrieben werden.

Abb. 52 zeigt ein Diagramm für Proteine *downstream* von TSS Vorhersagen für SigE und SigK.

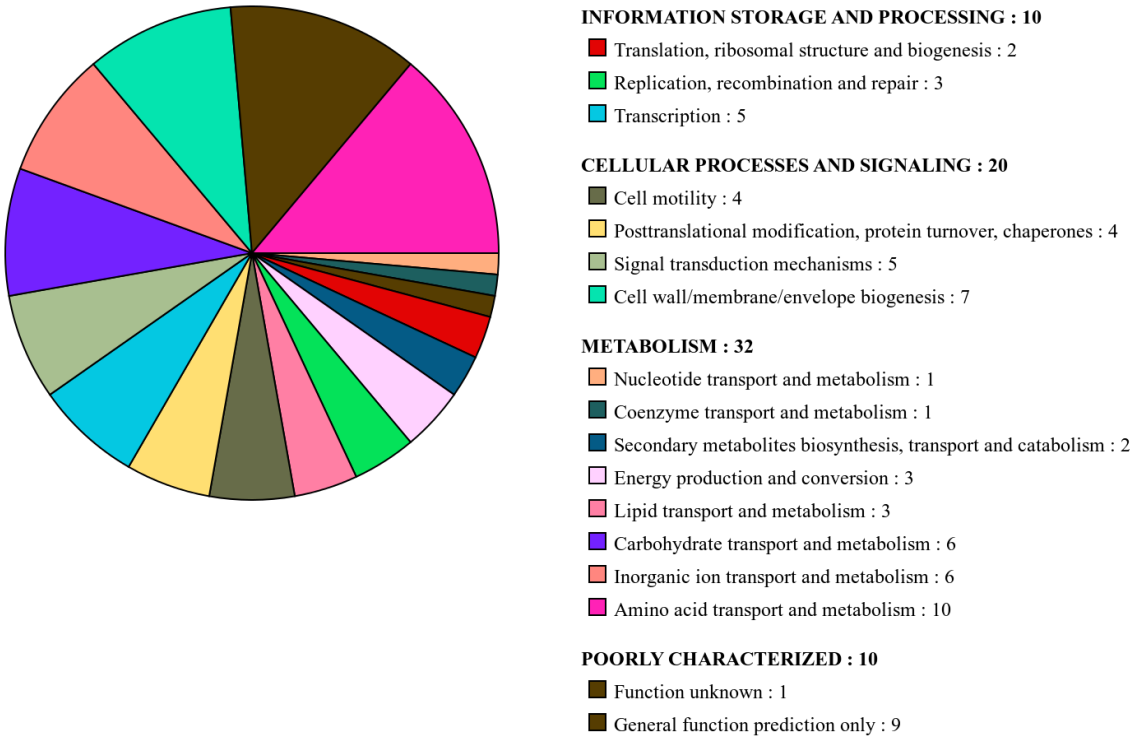

Abb. 52: Übersicht der COG Kategorien der ersten Proteine downstream von vorhergesagten TSS für SigE/SigK Promotoren
Der Großteil der klassifizierbaren Proteine besitzt Funktionen im Metabolismus und Transport von Kohlenhydraten, inorganischen Ionen und Aminosäuren. Desweiteren sind Funktionen aus der Zellwandsynthese und Transkriptionsregulation vertreten

Für die SigE/SigK Promotoren konnten insgesamt 149 Kandidaten vorhergesagt werden wovon 112 zu einem Protein zugeordnet werden konnten. Von diesen Proteinen konnten 72 einer COG Kategorie zugeordnet werden. Die COG Kategorie Metabolismus ist die am stärksten vertretene Kategorie, wovon die meisten Proteine im Kohlenstoffmetabolismus sowie im Transport und Metabolismus von Aminosäuren und anorganischen Ionen beteiligt sind. In der Kategorie intrazelluläre Prozesse sind vor allem Proteine der Membran und Zellwandsynthese sowie der Signaltransduktion vertreten. SigE und SigK sind Promotoren für Gene für die Sporulation, wobei SigE Gene für die frühe Mutterzelle und SigK Gene für die späte Vorspore kontrollieren (Sonenshein *et al.*, 2002; Eichenberger *et al.*, 2004). Eichenberger *et al.* benennen verschiedene Gene der Sporulation welche sie als SigE und SigK kontrolliert beschreiben. Innerhalb der MEME Vorhersagen konnten für SigK die Gene *cwlC*, *cotD* und *cotF* sowie *spoIIP*, *yngJ* und *comER* für SigE bestätigt werden. Betrachtet man die Annotationen der SigE/K regulierten Gene, sind 27 dieser Proteine an der Sporulation beteiligt, womit in etwa ein Viertel der regulierten Proteine zum SigE/SigK Regulon passen würden.

Die für den vollen TSS Kandidaten Datensatz erhaltenen Vorhersagen unterscheiden sich qualitativ nicht von denen des kurierten Datensatzes mit nur 1500 Kandidaten. Die Vorhersagen mit dem vollen Datensatz mit 3064 TSS Kandidaten liefert 221 *patterns* die mit MEME vorhergesagt wurden. Eine Suche mit dem kurierten Datensatz mit 1500 TSS Kandidaten liefert 87 *patterns*. Wendet man das in diesem Kapitel beschriebene Muster an um σ-Faktor *patterns* zu erhalten, führen beide Datensätze zu den gleichen Ergebnissen wobei sich die Reihenfolge in der sie erkannt werden unterscheidet und die Gewichtung der einzelnen konservierten Basen in den WebLogos geringfügig variiert. Die restlichen *patterns* in den Vorhersagen basieren auf wenigen Sequenzen im Vergleich zu den σ-Faktor *patterns*. Zusätzlich wurde geprüft, ob eine Standardsuche mit MEME, welche nach einer festen Anzahl von erwarteten *patterns* sucht, andere Ergebnisse liefert. Diese Suche wurde für 50 erwartete *patterns* durchgeführt in der Annahme, dass die erwarteten σ–Faktor *patterns* innerhalb der 50 Kandidaten auftauchen. Auch diese Suchen kamen zu vergleichbaren Ergebnissen und nur solche *patterns*, die auf einer geringen Anzahl von Sequenzen basieren, unterschieden sich und werden in den hier durchgeführten Analysen nicht betrachtet. Die entsprechenden Ergebnisse sind auf der Daten-CD im Verzeichnis Nimmersatt zu finden.

Für die Suche nach σ-Faktor Promotor *patterns* hat eine manuelle Kuration der Kandidaten sowie der Nimmersatt Algorithmus keine direkten Vorteile für die Sensitivität. Der Nimmersatt Algorithmus ist aber in der Lage mehr *patterns* zu erkennen und stellt sicher, dass solange

gesucht wird bis keine neuen *patterns* mehr gefunden werden können. Dies kann vorteilhaft sein wenn *patterns* mit nur wenigen Examplaren im Genom gesucht werden sollen.

8 Prophagenaktivitätsbestimmung

Phagen sind auf Bakterien und Archeaen spezialisierte Viren, welche sich mittels verschiedener Mechanismen in das Wirtsgenom integrieren können (Casjens, 2003). Die so integrierten Phagengenome können dann mit dem Wirtsgenom repliziert und vermehrt werden. Einen solchen Phagenbereich in einem Wirtsgenom nennt man Prophagen.

In Zusammenarbeit mit Robert Hertel wurde TraV für ein Projekt eingesetzt, bei dem das Potential von Prophagen aus *B. licheniformis* DSM13 untersucht wurde, DNA in Phagenpartikel zu verpacken. Der experimentelle Ansatz basiert auf einer Aufreinigung von Phagen Doppelstrang-DNA aus deren Partikeln und folgender NGS-Sequenzierung. Diese erhaltenen Phagen DNA Sequenzen wurden dann für ein *mapping* auf dem *B. licheniformis* DSM13 Genom verwendet. Mittels TraV wurden Bereiche mit erhöhter Aktivität aufgrund der Akkumulation der Phagen DNA Sequenzen im *B. licheniformis* DSM13 Genom identifiziert und von Robert Hertel auf ihre Korrelation mit annotierten Prophagenbereiche untersucht (Hertel *et al*. submitted).

Tabelle 36 listet die durch Robert Hertel identifizierten Prophagenregionen in *B. licheniformis* DSM13 auf.

Tabelle 36: Auflistung Phagenregionen in *B. licheniformis* DSM13 identifiziert durch Robert Hertel

Prophage*	Größe in Basen	Koordinaten im Genom
BLi_Pp1	11.177	927.299 – 938.595
BLi_Pp2	27.509	1.317.754 – 1.345.262
BLi_Pp3	41.566	1.422.556 – 1.464.174
BLi_Pp4	38.319	1.504.028 – 1.542.847
BLi_Pp5	10.524	2.855.587 – 2.866.209
BLi_Pp6	44.793	3.424.376 – 3.469.168
BLi_Pp7	21.733	4.155.490 – 4.177.258

*Im Rahmen der Auswertung der RNA-Seq Experimente wurde *B. licheniformis* DSM13 reannotiert was die Annotation der Prophagenregionen beinhaltet (Wiegand *et al.*, 2013).

Robert Hertel hat im Rahmen der Untersuchungen verschiedene Deletionsmutanten von *B. licheniformis* DSM13 erstellt und diese für die Phagenisolationen verwendet. Insgesamt wurden drei verschiedene Phagenisolate sequenziert, welche hier genauer betrachtet werden. Der erste Datensatz ist die Phagen DNA Isolation vom Ausgangsstamm DSM13 ohne Mutationen. Der zweite Datensatz ist eine Mutante bei der die BLi_Pp2 Region deletiert

wurde, da diese den PBSX-orthologen Phagen aus *B. licheniformis* DSM13 kodiert. PBSX-artige Phagen verpacken unspezifisch Teile des Wirtsgenoms. Dies führt im ersten Datensatz zu einem erkennbaren Hintergrund (Shingaki *et al.*, 2003). Durch die Deletion der BLi_Pp2 Region sollte dieser Hintergrund verschwinden. Der dritte Datensatz stammt von einer Doppelmutante, bei der die BLi_Pp2 sowie die BLi_Pp3 Region deletiert wurden. Diese Deletionsmutanten wurden auf dem Stamm MW3 von *B. licheniformis* durchgeführt. Der MW3 Stamm hat in BLi_Pp7 eine Deletion. Diese ist nötig um die Transformierbarkeit gegenüber DSM13 zu erhöhen, da in dieser Region Gene eines Restriktionssystems liegen (Waschkau *et al.*, 2008). Dies hat zu folge, dass Aktivitätsbetrachtungen der BLi_Pp7 Region in diesem Experiment nicht möglich sind da Teile des Prophagen deletiert wurden. Abb. 53, erstellt durch Robert Hertel, gibt einen Überblick über die Abdeckung des Genoms durch die Phagen DNA in den drei Datensätzen.

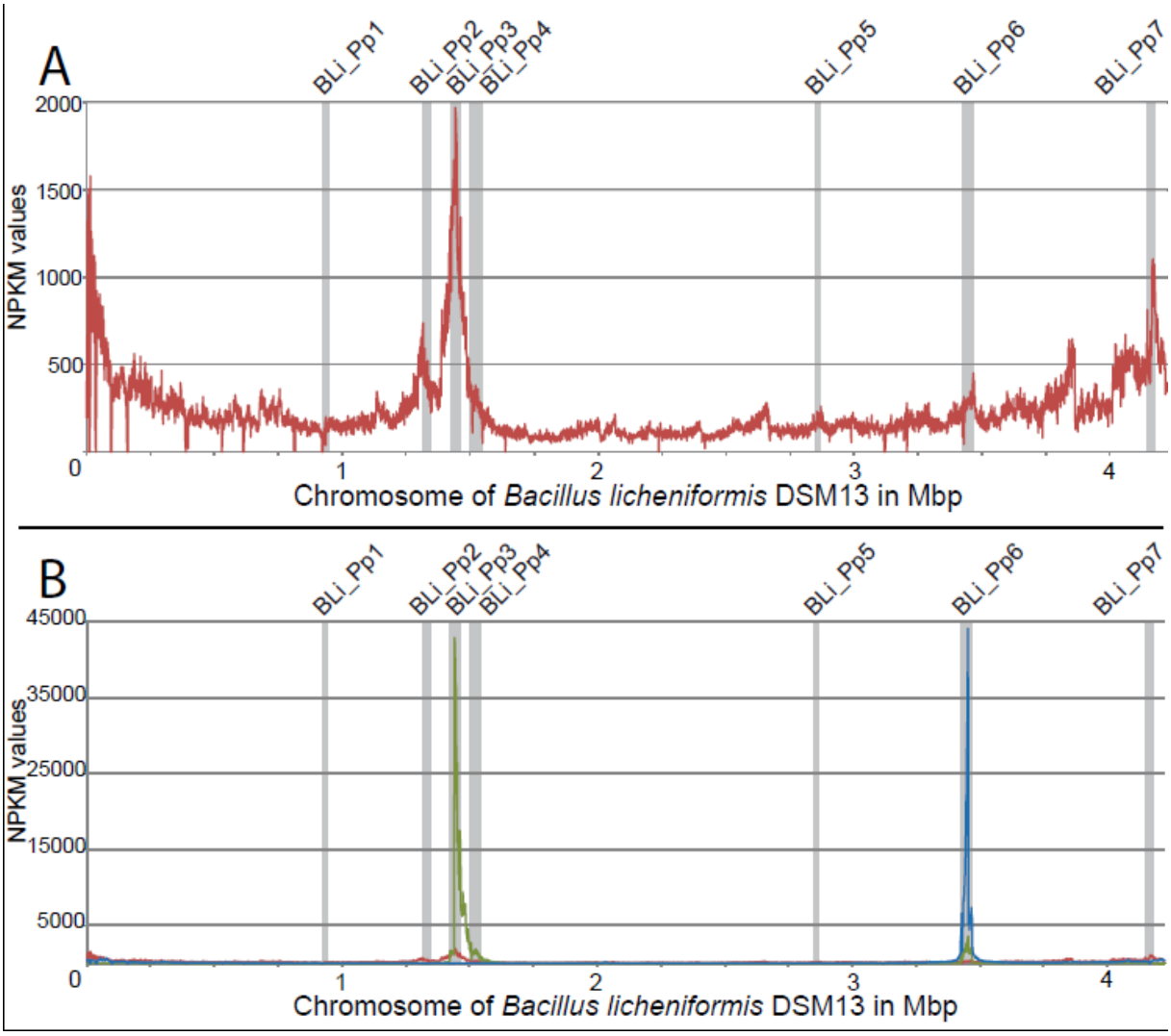

Abb. 53: Abdeckung des *B. licheniformis* DSM13 Genoms durch die Phagen DNA erstellt von Robert Hertel
Grafik A und der rote Graph in Grafik B zeigen die Abdeckung des Genoms wenn BLi_Pp2 intakt ist. Erkennbar ist die Abdeckung des gesamten Genoms durch diesen Phagen. Der grüne Graph in Grafik B zeigt die Abdeckung nach Deletion der BLi_Pp2 Region. Der blaue Graph in Grafik B zeigt die Abdeckung nach Deletion der BLi_Pp3 Region

Anhand dieser Prophagenbereiche wurde die durchschnittliche Basenabdeckung in den Prophagen und Nicht-Prophagenbereichen berechnet indem die Basenabdeckung der einzelnen Bereiche aufaddiert und durch die Länge der Bereiche geteilt wurde. Aus dem Verhältnis von durchschnittlicher Basenabdeckung im Prophagenbereich zum Nicht-Prophagenbereich wurde dann das *Signal-to-noise* Verhältnis errechnet.

Tabelle 37 gibt diese Verhältnisse in Prozent der gesamten Abdeckung über das Genom an.

Tabelle 37: Tabelle der durchschnittlichen Basenabdeckung von Phagen- und Nicht-Phagenbereichen in DSM13, MW3, MW3-BLi_ΔPp2 und MW3-ΔBLi_Pp2-ΔBLi_Pp3

Stamm	**Durchschnittliche Basenabdeckung im Prophagenbereich**	**Durchschnittliche Basenabdeckung im Nicht-Prophagenbereich**	***Signal-to-noise* Ratio (Prozentualer Anteil vom Noise an der Aktivität)**
DSM13	82.7	31.5	2.6 (38,1%)
MW3	187	79.7	2.3 (42,6%)
MW3-ΔBLi_Pp2	1447.2	27.2	53.2 (1,9%)
MW3-ΔBLi_Pp2-ΔBLi_Pp3 (1. Exp.)	1671.8	68.3	24.5 (4,1%)
MW3-ΔBLi_Pp2-ΔBLi_Pp3 (2. Exp.)	44.9	1.4	32.1 (3,1%)
MW3-ΔBLi_Pp2-ΔBLi_Pp3 (3. Exp.)	664.6	5.4	123.1 (0,8%)

Anhand von Tabelle 37 und Abb. 53 ist der Einfluss des PBSX-orthologen Phagen gut erkennbar, welcher in DSM13 und MW3 ca. 38,1% bzw. 42,6% der Aktivität der Nicht-Phagenbereiche bewirkt. Nach der Deletion dieses Phagen sinkt die Abdeckung der Nicht-Phagenbereiche auf ca. 2%. Die Deletion des BLi_Pp3 Prophagen zeigt keinen solchen Effekt, die Werte schwanken in den drei Deletionsexperimenten zwischen 4,1% und 0,8%.

9 Metatranskriptom einer Algenblüte aus der Nordsee

Unter einem Metatranskriptom versteht man die gesamte Transkriptionsleistung der Organismen in einem definierten Habitat. Metatranskriptome werden vor allem dann erstellt, wenn ein Organismus nicht einzeln kultivierbar oder deren Verhalten von der Gemeinschaft abhängig ist oder sein Transkriptionsverhalten stark von seinen Standortsbedingungen abhängig ist, welche nicht unter Laborbedingungen replizierbar sind (Moran *et al.*, 2013). Dies stellt den Metatranskriptomansatz vor drei Probleme. Erstens wird die Sequenzierleistung zwischen den verschiedenen Organismen aufgeteilt, so dass ein Metatranskriptom stets mehr Sequenzierleistung benötigt als eine vergleichbare Transkriptomsequenzierung für einen einzelnen Organismus. Zweitens können Standortbedingungen die Probengewinnung beeinträchtigen wenn z.B. nach Probenentnahme die Probe nicht sofort inaktiviert werden kann, so dass das Expressionsverhalten sich eventuell vom eigentlichen Standort unterscheidet oder bestimmte Chemikalien vom Probenstandort die Sequenzierung inhibieren (Tveit *et al.*, 2014). Genauso könnten Aufreinigungsmethoden einen Bias gegenüber bestimmten Organismengruppen verursachen. Drittens können ähnliche Sequenzen zwischen verschiedenen Organismen zu Fehlzuordnungen beim *mapping* führen.

In Zusammenarbeit mit Sonja Voget wurde ein Metatranskriptom einer Algenblüte aus der südlichen Nordsee ausgewertet (Voget *et al.*, 2014). Phylogenetische Analysen zeigten, dass Vertreter des RCA clusters 10-31,3% der bakteriellen Gemeinschaft der hier untersuchten Proben ausmachte (Wemheuer *et al.*, 2014). Das RCA *cluster* gehört zur weltweit in marinen Habitaten sehr abundanten Roseobacter Gruppe. Vertreter des RCA *clusters* stellen hierbei einen signifikanten Anteil im bakteriellen Plankton in temperaten bis subpolaren Gewässern (Wemheuer *et al.*, 2014). Es sollte daher die transkriptionelle Aktivität des α-Proteobakteriums *Planktomarina temperata* RCA23 (Giebel *et al.*, 2013), untersucht werden. Für einen Vergleich wurden neben *P. temperata* RCA23 zwei weitere Organismen für das *mapping* verwendet: *Candidatus Pelagibacter* ubique HTCC1062 (Giovannoni *et al.*, 2005), ein Vertreter der SAR11 *clade* und ebenfalls ein Isolat aus der Nordsee sowie HTCC2207 (Cho and Giovannoni, 2004), ein γ-Proteobakterium aus der SAR92 *clade*, welche ebenfalls an den analysierten Stationen sehr abundant waren.

Aus den Proben wurden insgesamt 78,042,122 einzelne cDNA *reads* mit einer Länge von 75 bis 100 Basen sequenziert: 24,879,579 für die Station außerhalb der Algenblüte, 26,176,832 für die Algenblüte in der Nacht und 26,985,711 für die Algenblüte am Tag. Diese Sequenzen wurden von Sonja Voget mit Trimmomatic (Bolger *et al.*, 2014) nach Qualität und Mindestlänge gefiltert und mittels Bowtie2 wurden *mappings* der drei Vergleichsorganismen erstellt. Für die *mappings* wurde Bowtie2 im *end-to-end* Modus verwendet, um nur *mappings*

zu erlauben, die über die gesamte *read* Länge passen. Die erhaltenen *mappings* wurden mit SAMtoTDS in das TraV Austauschformat konvertiert. Bei der Umwandlung wurde eine Mindestähnlichkeit zwischen *read* und Referenzsequenz von mindestens 90 Prozent als *cut-off* festgelegt. Die Proben teilen sich in drei Bedingungen auf, Beprobung der Algenblüte bei Nacht und bei Tag sowie eine Probe am Tag außerhalb der Algenblüte. Tabelle 38 gibt einen Überblick über die Anzahl an *mapped reads* und eine prozentuale Abdeckung der jeweiligen Genome durch die *mapped reads.* Diese prozentuale Abdeckung wurde anhand der abgedeckten Basen ermittelt. Eine Basenposition galt als abgedeckt, wenn mindestens ein *read* auf dem Plus- oder Minusstrang *mapped*. Die prozentuale Abdeckung berechnet sich aus dem Anteil der abgedeckten Basenpositionen zur Gesamtzahl der Basenpositionen im Genom.

Tabelle 38: Überblick über die Verteilung der *mapped reads* zwischen *P. temperata* RCA23, Cand. *P. ubique* HTCC1062 und HTCC2207

Bedingung	**Organismus**	***Mapped reads***	**Prozentuale Abdeckung des Genoms**	**Prozentualer Anteil der *mapped reads* an Sequenzierleistung**
Außerhalb Algenblüte	*P. temperata* RCA23	19.435	17,3	0,2%
	HTCC1062	16.947	17,9	
	HTCC2207	4.021	2,6	
Algenblüte, Nacht	*P. temperata* RCA23	543.596	93,4	2,3%
	HTCC1062	35.853	42,6	
	HTCC2207	27.689	34,1	
Algenblüte, Tag	*P. temperata* RCA23	1.222.858	94,6	5,3%
	HTCC1062	27.026	40,9	
	HTCC2207	179.429	89,1	

Anhand der Anzahl der *mapped reads* und der prozentualen Abdeckung des Genoms konnten Voget *et al.* zeigen dass *P.temperata* RCA23 innerhalb der Algenblüte anscheinend der aktivste Organismus ist. Anhand von mit TraV berechneten NPKM-Werten konnten Unterschiede in der Transkription verschiedener COG Kategorien auf der Gen Ebene gezeigt werden. Innerhalb der Phytoplanktonblüte in der Nacht waren z.B. Gene des Photosyntheseapperates überexprimiert. In der Nachtprobe konnten auch viele Transkripte nachgewiesen werden, die den RNA Polymerase sigma-32 Faktoren RpoH1 und RpoH2 nd der dazugehörigen Maschinerie an Stressproteinen zugeordnet werden konnten.

10 Diskussion

Das TraV Tool stellt einen neuen und erfolgsversprechenden Ansatz zur Auswertung von RNA-Seq Experimenten dar. Die Analysen, die innerhalb dieser Arbeit durchgeführt worden, zeigen die Vorhersagekraft des TraV-Ansatzes und die Möglichkeiten, die TraV bietet, um interessante *features* im Genom zu finden. Die Weboberfläche und Datenbank sind sehr performant, auch bei vielen Datensätzen, und sind gut bei Fernzugriff einsetzbar. Außerdem sind die erstellte Datenbank und die webbasierte Arbeitsoberfläche effiziente Grundlagen für die weitergehende Entwicklung von TraV in Bezug auf neue Algorithmen für die Beschreibung von neuen Klassen von *features* wie auch die Anbindung anderer *tools.* Beispiele für solche Anwendungen sind z.B. die Suche nach *patterns* und die Identifizierung von aktiven Prophagen. TraV ist derzeit begrenzt durch die Möglichkeiten der derzeitigen Sequenziertechnologien und Fortschritte in diesem Bereich werden genauere Analysen mit TraV ermöglichen.

10.1 TraV im Vergleich zu anderen *tools*

TraV ist in seiner Konzeption darauf ausgelegt, möglichst viele Datensätze gleichzeitig verarbeiten zu können. Diesem Anspruch wird es gerecht, indem *mappings* durch SAMtoTDS auf Basenaktivitäten abstrahiert werden. Indem die *read* Informationen verworfen werden, wird die Struktur der Daten erheblich vereinfacht und die Informationsmenge stark reduziert wie in 5.2 dargestellt. Viele vergleichbare *tools* verwerfen diese *read* Informationen nicht. Diese *tools* benutzen oft BAM formatierte *mapping*-Informationen direkt , wie z.B. Artemis (Carver *et al.*, 2012) oder SeqMonk (http://www.bioinformatics. babraham.ac.uk/projects/seqmonk). BAMs haben gegenüber SAMs den Vorteil, dass sie indexiert und sortiert sind (Li *et al.*, 2009). Jedoch reduziert dass nicht die Notwendigkeit, die einzelnen *read* Informationen zu *parsen* und zu interpretieren. Die Indexierung kann aber benutzt werden, um nur die Informationen für einen betrachteten Bereich zu laden. Das reduziert den Aufwand des *parsings* erheblich, negiert ihn aber nicht, da bei Veränderung des betrachteten Bereichs die Informationen nachgeladen werden müssen. Je nach Größe des betrachteten Bereichs und der Abdeckung des Genoms konnte beobachtet werden, dass dies zu spürbarem Rechenaufwand führt, was für ein interaktives *tool* sehr nachteilhaft ist. TraV kann aufgrund der reduzierten Größe der Datensätze diese komplett laden und muss so kein nachträgliches *parsing* betreiben. Damit ist bei TraV der Rechenaufwand, in Bezug auf die RNA-Seq Daten, linear von der Größe des betrachteten Bereichs abhängig. Dies ist besonders wichtig beim Vergleich vieler Datensätze miteinander, da sich mit der Menge der betrachteten Datensätze dieser Aufwand aufaddiert. Innerhalb der TraV *pipeline* muss ein SAM Datensatz nur einmalig komplett ausgelesen werden, nämlich bei der Konvertierung des SAM zu einem TDS. Damit ist der rechenaufwendige Schritt des *parsens* von der Betrachteroberfläche getrennt und beeinflusst diese somit auch nicht.

Andere *tools* wie IGV (Thorvaldsdóttir *et al.*, 2013) oder der UCSC Genome Browser (Kent *et al.*, 2002) können die sogenannten Wig und bigWig Formate verwenden, welche *read* Informationen verwerfen und im Aufbau dem TDS Format ähneln (Kent *et al.*, 2010). Das IGV *tool* ist in seiner Auslegung auf die Betrachtung von bereits bestehenden Annotationen ausgelegt. Der UCSC Genome Browser ist in der Lage viele verschiedene analytische *tools*, vor allem im Bereich der Eukaryoten, auf die Daten anzuwenden. In der Auslegung ist aber auch dieses *tool* fokussiert auf die Annotation und den Vergleich von Proteingenen. Strukturell ähnelt es aber dem TraV *tool* am ehesten aufgrund seiner zentralen Verwaltung der Daten und seiner Auslegung als *webtool*. Das *tool* VESPA (Peterson *et al.*, 2012) konzentriert sich auf die Entdeckung von fehlenden Annotationen in Genomen unter Verwendung von RNA-Seq Daten. Damit ähnelt es in seiner Methodik der TraV Suche nach *free transcripts* und benutzt UTR Bereiche zur Korrektur von Open Reading Frames (ORFs). Die Kombination von Expressionsdaten mit bestehenden Annotationen zur Aufdeckung von *features* wie UTRs oder *antisense transcripts* wird in diesen *tools* nicht geboten und muss manuell durchgeführt werden.

In seiner Auslegung ist TraV, mit seiner Fähigkeit viele Datensätze zu bearbeiten und zu kombinieren sowie der analytischen Methoden, welche sich auf die Identifizierung von genomischen *loci* mit möglichen Regulatoren konzentrieren, eine Ergänzung zu anderen *tools*, welche primär auf die Betrachtung der bestehenden Annotationen ausgelegt sind. TraV generiert in seinen Analysen keine klassifizierenden Aussagen, sondern generiert Kandidatenlisten für auffällige Bereiche im Genom basierend auf deren transkriptioneller Aktivität und dem genetischen Kontext. Die Strukturierung der Ergebnisse der analytischen Methoden folgt beschriebenen Standardformaten wie FASTA und GFF, so dass die Ergebnisse möglichst einfach als Eingabe für andere *tools* verwendet werden können. Sind also Betrachtungen der bereits bekannten Genannotationen mitsamt statistischer Expressionsanalysen von diesen Annotationen das Ziel, bieten andere *tools* wie z.B. baySeq (Hardcastle and Kelly, 2010) oder DESeq (Anders and Huber, 2010) bessere Optionen. Soll aber eine genomweite Suche nach Bereichen mit potentiellen, regulatorischen *features* stattfinden, bei der möglichst viele Bedingungen und damit Datensätze in die Suche einfließen sollen, bietet TraV die komfortabelsten Methoden und erstellt Ergebnisse, die einfach in anderen analytische *tools* wie z.B. einer Infernal (Nawrocki *et al.*, 2009) Suche mit den Rfam Kovarianzmodellen verwendet werden können.

10.2 Datenbank und Weboberfläche von TraV

Die in diesem Projekt anfallenden Datenmengen verlangen für eine effiziente Verwendung eine Datenbank, welche dynamische Zugriffe auf Teile der Datenbestände erlaubt. Die Notwendigkeit dieser dynamischen Zugriffe schließen eine dateibasierte Speicherung der Daten aus, da sie zeitaufwendiges *parsing* verlangen würde. Als Lösung wurde PostgreSQL gewählt, welches z.B. im ERGO Annotationssystem (Overbeek, 2003) eingesetzt wurde. Aufgrund voriger Arbeiten mit dem ERGO System lag bereits Erfahrung mit PostgreSQL vor. Eine alternative wäre MySQL, welches vergleichbar zu PostgreSQL ist.

Die entwickelte OmicsDatabase.jar Klassenbibliothek hat sich im Laufe ihrer Entwicklung von einem reinen *layer* für die Datenverwaltung für die TraV-Oberfläche zu einem vielseitigen Werkzeug entwickelt, das die Entwicklung weiterer, aufgabenspezifischer *tools* neben TraV erlaubt. Beispiele solcher *tools* sind z.B. Nimmersatt, welches für die Promotorvorhersagen verwendet wird. Die Berechnung der Abdeckung eines Genoms wie sie in Kapitel 8 und 9 stattfindet ist ebenfalls ein Beispiel für die Vielseitigkeit der OmicsDatabase Bibliothek. Solche Methoden sind gute Kandidaten für neue analytische Methoden für zukünftige Versionen von TraV.

Die TraV Oberfläche ist eine gut funktionierende Lösung für die Arbeit mit den RNA-Seq Daten. In verschiedenen Anwendungsbeispielen konnte TraV erfolgreich für die Bearbeitung von RNA-Seq Datensätzen eingesetzt werden (Wiegand *et al.*, 2013; Voget *et al.*, 2014). Auf einem dedizierten Server konnten erfolgreich insgesamt 80 Datensätze gleichzeitig von verschiedenen Personen geladen und bearbeitet werden. Obwohl sich TraV somit als effizients Analysetool erwiesen hat sind Verbesserungen vor allem in der *usability* möglich, da viele Aspekte der Oberfläche aufgrund der Entwicklungsgeschichte nicht optimal aufeinander abgestimmt sind. Viele Interaktionen mit den Daten geschehen derzeit über Seitenmenues im Browser wie z.B. der Zugriff auf Annotationsinformationen von Genen oder basengenaues Navigieren. Diese Aktionen könnten sehr viel effizienter durch Interaktionen mit den Graphen selber durchgeführt werden. Ein anderer Aspekt ist die Darstellungsweise von *locus tags* innerhalb des Graphen. So kann es passieren, dass *locus tags* bei großen Ausschnitten des Genoms überlappen. Lösungen könnten zum Beispiel eine automatische Staffelung, Rotation um 45 Grad oder ein Ausblenden der *locus tags* sein, abhängig von der verwendeten Ausschnittgröße des dargestellten Fensters.

Der momentan kritischste Aspekt für Verbesserung der *usability* ist das Einladen und Betrachten von nicht geschlossenen Genomen und ihrer Transkriptomdatensätze. Da TraV mit geschlossenen Abschnitten genomischer Informationen und deren Annotationen, den sogenannten *contigs* arbeitet, ergeben sich Probleme bei der Bearbeitung von nicht geschlossenen Genomen. Nicht geschlossene Genome besitzen in der Regel viele solcher

contigs und verfügen oft nicht über qualitative Annotationen. Derzeit müssen diese *contigs* stets einzeln geladen und betrachtet werden. Eine mögliche Lösung wäre, *contigs* und deren Datensätze in benutzerdefinierbare *scaffolds* zu vereinen, so dass mehrere *contigs* als ein artifizielles *super-contig* behandelt werden können. Das Einladen der Transkriptomdaten müsste dementsprechend auch angepasst werden. Derzeit muss über die Benutzeroberfläche für jedes *contig* ein Datensatz einzeln zugeladen werden. Mittels der OmicsDatabase Bibliothek ist es möglich, ein *tool* zu schreiben, dass mehrere Datensätze am Stück importiert und damit viel Interaktion mit der Benutzeroberfläche erspart. Diese Funktionalität sollte innerhalb der Oberfläche realisiert werden, sodass Benutzer in der Lage sind, die Struktur und Reihenfolge ihrer Datensätze frei und dynamisch zu bestimmen.

TraV ist auf die detaillierte Analyse von *features* in deren genomischen Kontexte ausgelegt. Eine Einbindung verschiedener Darstellungstools wie z.B. DNAPlotter (Carver *et al.*, 2009) oder Circos (Krzywinski *et al.*, 2009) wäre denkbar, um die Darstellungsmöglichkeiten von TraV zu erweitern. Somit könnten für Übersichtsdarstellungen von gesamten Genomen in einer zukünftigen TraV Version solche Darstellungsmethoden integriert werden.

10.3 Mapping

Das *mapping* der RNA-Seq Daten ist die Datengrundlage für alle Analysen und Darstellungen, die TraV generiert. Als solches ist eine korrekte Handhabung des *mappings* unerlässlich für die Verlässlichkeit der Vorhersagen von TraV. Diese Verlässlichkeit wird durch die konservativen Mindestanforderungen beim Prozessieren der *mappings* durch SAMtoTDS sichergestellt. Die Mindestanforderung von 98% Ähnlichkeit (ein *mismatch* in 50 Basen) soll sicherstellen, dass *reads,* wenn sie *mapped* sind, mit hoher Wahrscheinlichkeit von dieser Position im Genom stammen. Solche *reads* die als *unmapped* geführt werden, wurden mittels BLAST gegen die nt Datenbank von NCBI (ftp://ftp.ncbi.nlm.nih.gov/blast/db/) verglichen. Diese *reads* beinhalten Artefakte aus der Sequenzierchemie wie Adaptersequenzen oder Poly-Adenin *reads*, welche dazu führen dass der 98% *cut-off* unterschritten wird. Solche *reads* die nicht über solche Artefakte verfügten, konnten stets gegen *B. licheniformis* DSM13 *mapped* werden, wobei aber auch hier, wahrscheinlich aufgrund von Sequenzierfehlern, der 98% Ähnlichkeit nicht erfüllt wurde. Bei diesen Vergleichen gab es keinen *read* der einen signifikanteren Treffer gegen einen anderen Organismus als *B. licheniformis* DSM13 lieferte. Das größte Problem beim *mapping* stellen die *multimapped reads* dar, da sie oft zu *B. licheniformis* DSM13 passen aber aufgrund der Uneindeutigkeit nicht klar zu einem *locus* zugeordnet werden können. Die in TraV gewählte, konservative Handhabung ignoriert diese *mappings*. Das führt zu einem *mapping* Artefakt, nämlich dass repetetive Bereiche des Genoms keine Aktivität zeigen obwohl theoretisch

reads an die entsprechenden *loci* gepasst hätten. Dies kann zu Fehlinterpretationen bei Analysen führen. Für *B. licheniformis* DSM13 wurde daher mit GEMmappability (Marco-sola *et al.*, 2012) getestet, in welchem Ausmaß repetetive Bereiche vorliegen, die mit unserer Leselänge für die *reads* nicht eindeutig *mappable* sind. Für unsere Daten liegt der Anteil dieser Bereiche am Gesamtgenom bei 1,45% (Wiegand *et al.*, 2013). Wurtzel *et al*. haben in einem vergleichbaren Experiment auf *S. solfataricus* einen *cut-off* von ca. 90% verwendet während sie *multimapped reads* ebenfalls verwerfen. Die Menge an *unique mapped reads* ist mit ca. 7-15% ebenfalls vergleichbar.

Längere *reads* könnten das *mapping* verbessern, da mit größerer Leselänge die Wahrscheinlichkeit steigt, dass eindeutige Bereiche im Genom erreicht werden und so der *read* eindeutig *mapped* werden kann. Größere Leselänge wirken sich aber negativ auf die Sequenzierung von *small* RNAs aus, da ein Ausschluss von zu kurzen Fragmenten durch die Erstellung der Sequenzierlibrary dazu führt, dass diese eher verworfen werden (Li *et al.*, 2010). Eine andere und in TraV noch nicht ausgenutzte Möglichkeit stellen *paired-end reads* dar. Bei *paired-end reads* sind miteinander verbundene *reads,* welche von verschiedenen Enden eines Fragments stammen. Sollte einer der *read* Partner *multimapped* sein während der andere eindeutig *mapped* ist und liegen sie zusätzlich in passender Distanz (basierend auf der Länge des Fragments) zueinander könnte man den *multimapped read* basierend auf seiner Relation zum Partner eindeutig lokalisieren und so repetetive Bereiche besser abdecken. Diese Möglichkeit wurde bisher in TraV nicht realisiert und stellt eine interessante Möglichkeit zur Lösung des *multimap* Problems dar. Daher würde sich die *paired-end* Sequenzierung sich als Ergänzung anbieten. Eine mögliche Verbesserung von TraV wäre, wenn innerhalb des SAMtoTDS Konverters die Bereiche des Genoms, die *multimapped reads* beinhalten, automatisch identifiziert und mittels eines GFF mitausgegeben werden. Eventuell könnten diese Informationen auch im TDS Format mitgeführt werden und innerhalb der TraV-Graphen speziell markiert werden. Dies würde die *multimap* Problematik nicht lösen aber zumindest die manuelle Betrachtung dieser entscheident vereinfachen.

Allen bisher verfügbaren Sequenziertechnologien ist gemein, dass sie Transkripte schären müssen und daher nur Stücke eines Transkripts sequenzieren können (Thorstenson *et al.*, 1998). Da die Teilstücke nicht mehr zu ihrem Transkript zurückverfolgt werden können, geht die Information über die Länge des Transkripts verloren. Da die Termination der Transkription oft nicht vollständig ist, kann es vorkommen dass Transkription in benachbarte Operons hineinläuft (Lewin, 2008). Was in der Zelle zwei eigenständige Transkripte sind, wäre im *mapping* ein geschlossener Bereich, der auf den ersten Blick nur ein Transkript suggeriert (siehe 10.4 und Abb. 54 für ein Beispiel für so eine Überlappung). Neuere Sequenziertechnologie wie PacBio (Paprotka *et al.*, 2012; Thürmer, 2014) bieten die

Möglichkeit, gesamte Transkripte in einem Stück zu sequenzieren. Sollte diese Sequenziertechnologie für RNA-Seq anwendbar werden, würde sie viele Vorteile und Möglichkeiten bieten, wie z.B. die Überbrückung von repetetiven Bereichen und die Aufklärung von Operonstrukturen in Genomen.

10.4 Analysemethoden und Vorhersagen von TraV für B. licheniformis DSM13

Die Analysemethoden, die TraV bietet, unterscheiden sich in vielen Aspekten von den Analysemethoden die andere *tools* zur Auswertung von RNA-Seq Daten bieten. Die meisten *tools* konzentrieren sich auf die Auswertung der Aktivität von bereits bestehenden Annotationen im Genom. In der Regel bedeutet das einen Fokus auf proteinkodierende *features*. *Tools* wie SeqMonk haben die Möglichkeit GFFs einzuladen und so die Reichweite der betrachtbaren *features* zu erweitern. Diese *tools* bieten jedoch keine Methoden um *features* zu finden, die sich in Abhängigkeit von Annotationen definieren, wie z.B. UTRs oder *free transcripts* und *antisense transcripts*. TraV wurde in seiner Konzeption auf die Analyse dieser *features* ausgelegt. Die Analysemethoden für 5‘ und 3‘ UTRs bieten die gleiche Effizienz wie aufwendigere Labormethoden zur Bestimmung von Transkriptenenden mit RACE(*rapid amplification of cDNA ends*)(Frohman *et al.*, 1988). So konnte zum Beispiel der Transkriptionsstartpunkt (TSS) des *thiD* Gens mittels RNA-Seq Vorhersage so genau bestimmt werden wie mittels 5’RACE (Denschlag, 2010). Auf diese Weise sind genomweite Bestimmungen der UTRs sowie von deren TSS mit guter Präzision möglich. Die *free transcript* suche erlaubt eine effiziente Suche nach nicht annotatierten Genen im Genom (siehe *bsrG* Beispiel, 6.5). Als Labormethoden für so eine Suche kämen aufwendigere, genomweite *microarrays* oder Proteinsequenzierung in Frage (Nicolas *et al.*, 2012; Ziady and Kinter, 2009).

Die Analysemethoden von TraV konnten erfolgreich auf die RNA-Seq Daten von *B. licheniformis* DSM13 angewendet werden (Wiegand *et al.*, 2013). Die Analyse wird jedoch durch leichte Asynchronität der Replikate in den frühen Phasen (Phase 1 und 2) der Fermentation erschwert. Diese Asynchronität bedingt sich aus der Beprobung der Versuchsfermenter anhand der Fermentationsparameter (siehe Kapitel 6). Die Messung dieser Parameter ist verglichen mit optischen Dichtemessungen der Kultur relativ ungenau aber notwendig, da das in der Fermentation verwendete Medium eine optische Dichtemessung nicht zulässt (Wiegand *et al.*, 2013). Die Folge ist dass die Replikate sich oft in den frühen Phasen der Fermentation unterscheiden und dies zu unterschiedlichen Aussagen in Bezug auf die Verhältnisse der Phasen 1 und 2 führt.

Mittels des Vergleichs von NPKM-Werten für die Gene von *B. licheniformis* DSM13 kann am Beispiel der *hag* und *spoIVA* Gene sowie des Inositol Operons gezeigt werden, dass mittels

RNA-Seq das Expressionsverhalten der Gene in Relation zu Wachstumsphase und dem Medium betrachtet werden kann. Das *hag* Gen kodiert für eine Strukturkompenente des Flagellums und wird im Verlauf der Fermentation immer mehr herunterreguliert. Da *hag* von SigD kontrolliert wird, ist eine Betrachtung des Expressionsverhaltens eventuell aufschlussreich für das Expressionsverhalten von *hag* selber. Bei Betrachtung der Aktivität von *sigD* zeigt sich, dass *B. licheniformis* unter den Fermentationsbedingungen anscheinend keine Motilität ausbildet und somit der Grund ist für die Abnahme der Aktivität von *hag*. Auf diese Weise lassen sich anhand der RNA-Seq Daten Schlüsse über die Regulation mancher Gene in Abhängigkeit vom Medium ziehen wobei wahrscheinlich nicht alle Genaktivitäten so direkte Schlüsse erlauben aufgrund komplexerer Regulationsnetzwerke wie z.B. bei den *rap*-Genen (siehe 10.4.3).

Am Expressionsverhalten von *spoIVA* kann die Regulation in Abhängigkeit von der Wachstumsphase gezeigt werden. SpoIVA ist eine Kompenente des Sporenmantels und sollte entsprechend nur während der stationären Phase gebildet werden (McKenney *et al.*, 2013). Die Expressionsprofile und NPKM-Werte von *spoIVA* bestätigen dies. Anhand des Inositol Operons lässt sich die Reaktion von *B. licheniformis* auf neue Nahrungsquellen im Medium zeigen. Inositol ist eine häufige Kohlestoffverbindung im Boden und kann von vielen Mikroorganismen verwendet werden (Yoshida *et al.*, 1997). Daher ist es für *B. licheniformis* von Vorteil, solche C-Quellen schnell aufzubrauchen bevor Konkurrenten sie verwenden können. Das Expressionsverhalten des Inositol Operons zeigt eine starke Aktvität in Phase 2, wo zuvor in Phase 1 noch keine Aktivität sichtbar ist. Diese Aktivität sinkt in den folgenden Phasen auf ein Minimum ab. Ob dies durch Katabolitrepression oder durch Aufbrauchen des Inositols im Medium geschieht, ist nicht klar.

Die 5‘-, 3’UTR Suche sowie die Identifikation von *free transcripts* haben verlässliche Ergebnisse geliefert, was die geringe Anzahl an manuellen Korrekturen zeigen (siehe Kapitel 6). Außerdem konnten 34 der 47 sRNAfinder Vorhersagen aus meiner Diplomarbeit mit Kandidaten aus den TraV Listen korreliert werden. Die fehlenden 13 Vorhersagen konnten durch TraV nicht erkannt werden, da das Expressionsverhalten derzeit diese *features* vor den derzeitigen Analysemethoden maskiert. Abb. 54 zeigt einen solchen Fall, in dem zwei Transkripte überlappen und so die 5’UTR mit einem *riboswitch* maskiert.

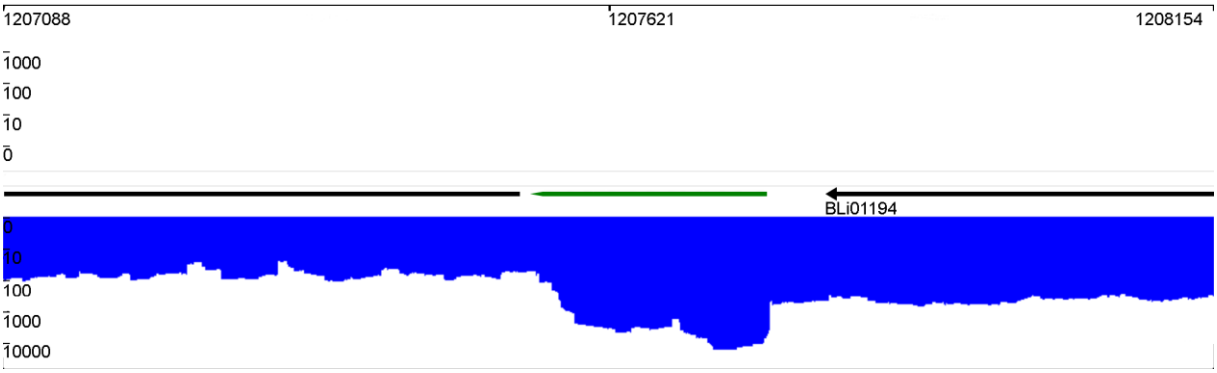

Abb. 54: Maskierung einer 5'UTR durch auslaufendes Transkript vom vorigen Gen (BLi01194)
Der grüne Pfeil markiert den Bereich mit dem *riboswitch.* Derzeitige Analysemethoden in TraV können solche maskierten 5'UTRs nicht identifizieren

Solche maskierten *features* sind ein bestehendes Problem für die TraV Analysemethoden, da sie in ihrem Erscheinungsbild den intergenischen Bereichen von polycistronischen Operons ähneln. Zur Erkennung dieser *features* sind heuristische Ansätze notwendig, die Länge der intergenischen Bereiche und das Vorhandensein von möglichen TSS als Merkmale für mögliche RNA-*features* verwenden.

Im Vergleich zu anderen Methoden, die sich mit der Vorhersage von 5' und 3' UTRs befassen, wie z.B. RACE und *microarrays*, ist der experimentelle Aufwand bei der RNA-Seq gering. Nicolas *et al.* (Nicolas *et al.*, 2012) erreichen mit genomweiten *microarrays* vergleichbare Ausbeuten wie Wiegand *et al*. mit der RNA-Seq Methode (Wiegand *et al.*, 2013). Nicolas *et al.* haben dabei 104 verschiedene Anzuchtsbedingungen für *Bacillus subtilis* betrachtet und für diese Bedingungen *microarrays* mit einer Auflösung von 22 Basen auf der aufgereinigten RNA durchgeführt. Durch Nicolas *et al.* wurden 676 5'UTRs länger als 50 Basen gefunden während wir mit der RNA-Seq und TraV Analyse 859 5'UTRs solcher UTRs idenfizieren konnten. Bei den 3'UTRs konnten mit der RNA-Seq jedoch weitaus mehr Kandidaten identifiziert werden als mittels der *microarrays*, nämlich 883 3'UTRs länger als 50 Basen mit der RNA-Seq Methode gebenüber 249 Kandidaten mittels der *microarrays.* Im Bereich der *free transcripts* (in Nicolas *et al.* als Indep bezeichnet) größer als 50 Basen konnten mit der RNA-Seq Methode 415 Kandidaten vorhergesagt werden, mittels *microarray* 153 Kandidaten. Nicolas *et al.* haben ebenfalls über die Korrelation von *upshifts* der Aktivität Vorhersagen für Transkriptionsstarts gemacht. Dabei wurden 3242 Kandidaten für TSS gefunden, was vergleichbar mit den 3064 Kandidaten aus den RNA-Seq basierten Vorhersagen ist.

Die TSS und *antisense transcripts* Suchen funktionieren und liefern sinnvolle Kandidaten. Sie besitzen aber noch Verbesserungspotential, da viele Kandidaten nach manueller Kuration verworfen wurden. Beide Methoden könnten über Heuristiken verbessert werden, die zusätzlich auf die Kandidatenlisten angewandt werden. Im Falle der TSS wäre dies z.B.

die Korrelation mit 5'UTRs oder eine Untersuchung auf den AT-Gehalt um die -10 Promotorbox, da diese in der Regel AT-reich ist um so das öffnen der DNA zu erleichtern (Feklistov and Darst, 2011). Desweiteren könnten Abgleiche mit bekannten Schnittstellen für Restriktionsenzyme gemacht werden, um so Prozessierungsstellen von TSS zu unterscheiden. Die *antisense transcripts* könnten über Längen und NPKM *cut-offs* auf wahrscheinliche Kandidaten reduziert werden und auf Korrelation mit regulatorischen Elementen, wie Promotorbindestellen und Shine-Dalgarno Sequenzen auf dem Gegenstrang überprüft werden. Weitere Hinweise auf *antisense transcripts* können in den *multimapped reads* liegen, wenn nämlich Bereiche einer in *trans* wirkenden *antisense* RNA zu einem anderen Bereich des Genoms komplementär sind. Im Vergleich zu anderen Methoden für die Vorhersage von TSS und *antisense transcripts* liefert die RNA-Seq basierte Vorhersage mit TraV vergleichbare Ergebnisse zu anderen Anwendungen der RNA-Seq Methode zu diesem Zweck (Sharma *et al.*, 2010). Sharma *et al.* konnten 1907 TSS Kandidaten mittels 454 Sequenzierung in *Helicobacter pylori* identifizieren. TraV generiert derzeit wahrscheinlich noch zu viele falsch positive Treffer, da die Analysemethoden wie oben beschrieben noch verbessert werden können.

Das *merging* Verfahren erlaubt schnelle und verlässliche Identifikation und Vergleich von Kandidaten in vielen Datensätzen. Da im *merged* Datensatz die Aktivitäten aller verglichenen Bedingungen einfließen, werden auch solche Kandidaten erkannt die nur in einem Teil der Datensätze aktiv sind. In der Berechnung der Aktivitätswerte in den einzelnen Datensätzen wird für diese Kandidaten in den nicht aktiven Bedingungen keine Aktivität gefunden. Dies erlaubt eine schnelle Identifizierung von bedingungsspezifischen *features* indem solche Kandidaten gesucht werden können, die nur in einem Teil der Bedingungen Aktivität zeigen. Der Nachteil diese Methode ist, dass Kandidaten derzeit immer in ihrer größten Ausdehnung betrachtet werden. Wenn unter unterschiedlichen Bedingungen verschiedene Längen eines *features* vorliegen (z.B. durch alternative TSS) wird dies derzeitig nicht berücksichtigt.

10.4.1 *Riboswitch* Vorhersagen

Mit den Kandidaten der TraV Analysen (siehe Kapitel 6) wurden Rfam Suchen durchgeführt um so spezifische regulatorische RNAs zu identifizieren. Exemplarisch wurden die *thiamine-pyrophosphate* (TPP), *S-adenosylmethionine* (SAM) und *flavin mononucleotide* (FMN) *riboswitches* betrachtet (siehe Kapitel 6.1, 6.2 und 6.3). Die Anzahl der gefundenen Vertreter dieser *riboswitches* entspricht den Vorhersagen aus meiner Diplomarbeit (Dietrich, 2009). Alle vorhergesagten *riboswitches* liegen in der Nachbarschaft von Genen, die mit dem Metabolismus oder Transport der vom *riboswitch* erkannten Metabolite assoziiert sind. Beim Abgleich der Struktur der Kovarianzmodell mit den Sequenzen der vorhergesagten *riboswitches* zeigen sich gute Übereinstimmungen mit den Modellen. In einzelnen Bereichen

der *riboswitches* deuten sich jedoch Variationen zum Modellkonsensus an, welche in den SAM *riboswitches* besonders auffällig sind. Diese Variationen könnten die Ursache für die unterschiedliche Regulationsstärke der *riboswitches* sein, welche sich in den unterschiedlichen Verhältnissen der Expressionstärke zwischen *riboswitch* und Genen andeutet. Ein Beispiel für eine solche Veränderung des Regulationsverhaltens bei *riboswitches* durch Veränderung der Sequenz sind ROSE Elemente, wo Punktmutationen dazu genutzt wurden, das Regulationsverhalten zu verändern (Chowdhury *et al.*, 2003). Anhand der Expressionsverhältnisse kann man abschätzen, unter welchen Bedingungen die *riboswitches* die Transkription erlauben und unter welchen sie die Transkription unterbinden, wobei dies aber nur relativ zwischen den betrachteten Bedingungen stattfinden kann. Auffällig ist, dass verschiedene *riboswitches* der gleichen Famile unterschiedliche Verhältnisse aufweisen. Dies ist ein Hinweis darauf, dass die *riboswitches* selber unterschiedlich starke Einflüsse auf die Transkriptionsrate haben. Dies könnte an den in den Rfam Vergleichen gefundenen Variationen in den Strukturen der *riboswitches* liegen. Alle in dieser Arbeit betrachteten *riboswitches* sind anscheinend transkriptionelle Regulatoren was gut mit den Aussagen des Reviews von Nudler und Mironov für Gram-positive Bakterien übereinstimmt (Nudler and Mironov, 2004). Das heißt, sie unterbrechen die Transkription durch eine Terminatorstruktur, die bei Bindung des von ihnen erkannten Metabolits gebildet wird. Fehlt der Metabolit, bildet sich eine Antiterminatorstruktur aus, welche die Bildung der Terminatorstruktur unterbindet und somit die Transkription der Gene erlaubt. Zusätzlich kann es *riboswitches* geben, die über Anti-Antiterminatorstrukturen verfügen. Diese verhindern die Ausbildung der Antiterminatorstruktur und bilden somit eine zusätzliche Ebene der Regulation. Anhand von Vergleichen des FMN *riboswitch* Modells mit Untersuchungen von Vitreschak *et al.* (Vitreschak *et al.*, 2002) lässt sich zeigen, dass die Rfam Modelle nur Teile der funktionellen RNA-Struktur beschreiben und funktionelle Bereiche wie zum Beispiel die Terminator *stem-loops* nicht Teil des Modells sind. Anhand der TransTermHP Vorhersagen und den Expressionsprofilen der *riboswitch* Kandidaten lässt sich bei vielen der vorhergesagten *riboswitches* die Anwesenheit von Terminatoren vermuten.

In den hier betrachteten Daten weisen die TPP-*riboswitches* in den Verhältnissen der Expressionsstärke geringe Unterschiede im Verhalten auf. Allgemein scheinen die TPP-*riboswitches* während des exponentiellen Wachstums die Transkription zu erlauben während sie in den anderen Phasen stärker reprimieren. Die vorhergesagten *riboswitch* Strukturen deuten ebenfalls auf ein vergleichbares Verhalten hin, da sie nur geringe Unterschiede zwischen den vorhergesagten Kandidaten aufweisen. Keiner der *riboswitches* scheint die Transkription der von ihm kontrollierten Gene komplett zu unterdrücken, was an den NPKM-Werten und den Expressionsprofilen ersichtlich ist.

Die SAM-*riboswitches* weisen im Vergleich zu den TPP-*riboswitches* sehr viel höhere Unterschiede in den Verhältnissen zwischen *riboswitch*- und Genexpression auf (siehe Tabelle 20). Die vorhergesagten *riboswitch* Strukturen weisen ebenfalls starke Unterschiede in einigen *stemloops* auf, was ein Hinweis auf unterschiedlich stabile Strukturen und damit Regulation sein kann. Von den SAM kontrollierten Operons sind vor allem *cysH1P1/sat/cysC, metK* und *metQ2N2P2* aktiv. Diese Operons sind beteiligt an der Synthese von *cystheine* und *S-adenosylmethionine,* sowie dem Transport von *methionine.* Die Operons *mtnKA* und *mtnWBD* sind an der Wiedergewinnung von *methionine* beteiligt und zeigen wenig Aktivität (siehe Abb. 27). Die *methionine* Transporter *metQ1N1P1* und BLi03178 zeigen ebenfalls kaum transkriptionelle Aktivität. Warum diese anscheinend inaktiv sind und der *metQ2N2P2* Transporter Transkripte aufweist, ist unbekannt. Denkbare Ansätze zur Aufklärung wären hier z.B. Modifikationen der variablen Bereiche in den *riboswitches* um Veränderungen des Regulationsverhaltens zu prüfen und zu vergleichen oder Deletionen in den vorhergesagten Terminatorstrukturen der *riboswitches* um so den regulatorischen Effekt zu unterdrücken. Die Umwandlung von *cystheine* zu *methionine* scheint ebenfalls stark herunterreguliert, erkennbar an den Operons *yxjG, metIC* und *yitJ/metH.* Aus diesem Expressionsverhalten lässt sich schließen, dass *B. licheniformis* DSM13 innerhalb der Fermentation *methionine* nicht aus *cystheine* synthetisiert und es stattdessen aus dem Medium bezieht. Das Operon *yitJ/metH* weist eine lückenhafte Abdeckung des Operons auf. Da für diese Bereiche keine Hinweise auf Transkriptionsstarts in den dRNA-Seq Daten zu finden sind, ist zu vermuten, dass diese Lücken durch die Prozessierung durch RNAsen entstehen. Aufgrund der Unvollständigkeit der Abdeckung dieses Operons wird hier angenommen, dass die betroffenen Transkripte nur im geringen Maße translatiert werden. Auffällig im Vergleich zu *B. subtilis* ist das *cysH1P1/sat/cysC* Operon. In *B. subtilis* ist dieses Operon anders strukturiert und beinhaltet *cysG/sirBC* wobei die letzten beiden Gene durch mRNA *processing* separat kontrolliert werden (Mansilla *et al.*, 2000). Die hier gezeigten Daten legen nahe, dass in *B. licheniformis* DSM13 *cysG/sirBC* im Unterschied zu *B. subtilis* ein eigenständiges Operon mit eigenem Promotor ist.

Die FMN-*riboswitches* verhalten sich ähnlich wie die TPP-*riboswitches.* Die Verhältnisse von *riboswitch* Expression zu den kontrollierten Genen suggerieren, dass die FMN-*riboswitches* während der exponentiellen Wachstumsphasen die Transkription ihrer Gene erlauben, wobei aber in den anderen Phasen die Transkription nicht komplett unterdrückt wird. Das Regulationsverhalten der FMN-*riboswitches* deutet auf einen erhöhten *riboflavin* Bedarf während des exponentiellen Wachstums hin. Interessant ist das *ribTHAED* Operon, welches für *ribT* und *ribH* während der stationären Phasen alternative Transkripte aufzuweisen scheint. Bei *ribT* und *ribH* gibt es TSS Kandidaten, wobei für den TSS Kandidaten von *ribT*

ein möglicher SigA Promotor erkennbar ist während *ribH* keinen bekannten Promotor aufweist. Zwar liegt ein SigH -10 *pattern* 5 Basen vor dem TSS, es gibt jedoch kein -35 *pattern.* Eventuell liegt hier eine regulatorische Struktur oder eine Prozessierungsstelle vor. Zur Klärung der Bedeutung dieser alternativen Promotoren für den Organismus sind weiterführende Experimente nötig. Mögliche Ansatzpunkte wären zum Beispiel Deletionsmutanten der Promotorregionen oder *in trans* Versuche durch den Einbau der Promotoren in ein Reportersystem. Auf jeden Fall ist es ein auffälliges Merkmal und ein Hinweis auf komplexe *multi-layer* Regulation.

Eine auffällige Signatur für die hier beschriebenen *riboswitches* ist die erhöhte Basenaktivität zwischen TSS und dem ersten Gen des Transkripts. Diese Erhöhung ist wahrscheinlich das Resultat der höheren Stabilität von gefalteter RNA gebenüber ungefalteter mRNA (Shahbabian *et al.*, 2009) sowie deren spezifischer regulatorischer Einfluss auf die Transkription, welche die Expression der nachfolgenden Gene reduziert. Die Vorhersagen zeigen, dass die RNA-Seq basierte Suche nach solchen regulatorischen Elementen erfolgreich war und dank der Verfügbarkeit der Expressionsininformationen einen besseren Einblick in deren Verhalten und Interaktion mit ihrem Kontext ermöglichen als dies reine *in silico* Vorhersagen erlauben würden. Da über die Suche nach Signaturen im Expressionsverhalten auch neue regulatorische Elemente vorhergesagt werden können, für die es bisher keine Modelle gibt, kann die RNA-Seq Methode einen Beitrag für die Verbesserung von Modellvorhersagen leisten. Solche Vorhersagen verlangen dann aber experimentelle Bestätigung.

10.4.2 *bsrG* Toxin/Anti-toxin Systeme

Neben der Vorhersage von regulatorischen *features* konnten Kandidaten für funktionelle RNAs wie z.B. das *bsrG* Toxin/Anti-toxin System gefunden werden. Auffällig ist, dass die Verhältnisse der Expressionsstärke zwischen den *bsrG* und SR4 Paaren in *B. licheniformis* DSM13 stark von den in der Literatur beschriebenen Verhältnissen für *B. subtilis* abweichen. Jahn *et al.* (Jahn *et al.*, 2012; Jahn and Brantl, 2013) beschreiben den SR4 Promotor um 6-10 mal stärker als den *bsrG* Promotor. Zwei der hier gefundenen *bsrG*/SR4 Kandidaten (*bsrG*/SR4_1 und *bsrG*/SR4_3) kehren dieses Verhältnis nahezu um. Geht man von den beschriebenen Längen der Transkripte aus und betrachtet die Annotationen, kann man eine Verwechslung der Transkripte ausschließen. Das *bsrG*/SR4_2 Paar zeigt ein weniger extremes Verhältnis, wo in einigen Fällen der SR4 Kandidat mehr Transkripte aufweist als der *bsrG* Kandidat, wobei aber auch hier das von Jahn *et al.* beschriebene Verhältnis nicht erreicht wird. Da die *bsrG* RNA das Toxin-Gen trägt, müsste man erwarten, dass die Kulturen bei den hier beobachteten Verhältnissen absterben. Da dies nicht der Fall ist, lässt sich vermuten, dass andere Mechanismen sich hier auswirken.

Die erste Möglichkeit ist, dass die hier mit dem *bsrG* Rfam Modell gefundenen Kandidaten keine *bsrG*/SR4 Paare sondern andere mRNA/sRNA Paare sind, welche einen ähnlichen Regulationsmechnismus verwenden. Diese Möglichkeit liegt nahe, da in *B. subtilis* nur ein *bsrG*/SR4 Paar beschrieben ist und in Sequenzhomologievergleichen in *B. licheniformis* kein *bsrG*/SR4 Kandidat gefunden werden konnte (Jahn *et al.*, 2012). Die zweite Möglichkeit wäre, dass es sich tatsächlich um *bsrG*/SR4 Homologe handelt, diese aber im Vergleich zu *B. subtilis* unter der Kontrolle weiterer Regulationsmechnismen stehen, sodass die höhere Expressionsrate von *bsrG* gegenüber SR4 nicht automatisch zum Zelltod führt. Die tatsächliche Natur dieser Kandidaten lässt sich aber derzeit nur experimentell bestimmen.

10.4.3 Response regulator aspartate phosphatases

Zusätzlich zu den funktionellen und regulatorischen RNAs wurden Expressionsvergleiche von proteinkodierenden Genen gemacht. Die in dieser Arbeit betrachteten *response regulator aspartate phosphatases* (*rap*-Gene) zeigen sehr gut die Möglichkeiten, die die RNA-Seq für die Aufklärung des Expressionsverhaltens von Regulatorproteinen bietet. Die *phr*-Gene, welche die *rap*-Gene regulieren, sind aufgrund ihrer Größe schwierig zu annotieren. Die Expressionsprofile zeigen sehr gut, ob ein *phr* existiert und wo dieses *phr*-Gen liegt. Außerdem bietet die RNA-Seq die Möglichkeit, den zugehörigen Promotor zu den *phr*-Genen zu identifizieren. Die hier durchgeführten Analysen zeigen, dass die *rap*- und *phr*-Gene anscheinend nur zum Teil von den gleichen σ-Faktoren kontrolliert werden wie in *B. subtilis* beschrieben (Mcquade *et al.*, 2001; Jarmer *et al.*, 2001). Es zeigen sich bei einigen *rap/phr* Genen Abweichungen oder Ergänzungen zum *B. subtilis* Verhalten. Das *rapA* Gen liegt doppelt vor, wobei aber unter den untersuchten Bedingungen nur eines der *rapA*-Gene aktiv zu sein scheint, nämlich *rapA2.* Interessanterweise sind die *phrA1* und *phrA2*-Gene aktiv und unterstehen beide der Kontrolle von SigA, wobei phrA2 einen SigH Promotor aufweist, welcher aber nach den Expressionsprofilen her nicht aktiv zu sein scheint, was mit den Aussagen von Mcquade *et al.* übereinstimmt. Das *rapH* Gen, welches wie *rapA1*, in dessen Nachbarschaft es liegt, ist unter den Versuchsbedingungen inaktiv. Es konnte außerdem keine Aktivität für ein *phrH* Gen festgestellt werden. Das *rapI* Gen wird anscheinend über das vorhergehende Gen *yhaR* mitabgelesen, obwohl ein potentieller SigA Promotor vorliegt. Für das *phrI*-Gen konnte kein eindeutiger Promotor gefunden werden. Mcquade *et al.* beschreiben für *phrI* in *B. subtilis* einen SigA und SigH Promotor. In *B. licheniformis* DSM13 konnte kein Hinweis auf solche Promotoren vor dem zugehörigen TSS gefunden werden. Die TSS besitzt in der *upstream* Region jedoch Sequenzenabschnitte, die einem SigE Promotor ähneln. Das *phrK*-Gen wird anscheinend von SigA kontrolliert, wobei es jedoch Sequenzen vor dem TSS gibt, die den SigH Konsensus erfüllen. Interessanterweise scheint dieser *locus* keine transkriptionelle Aktivität hervorzurufen, nur der SigA Promotor scheint aktiv zu sein. Mcquade *et al.* zeigen dass in

B. subtilis phrK über einen aktiven SigH Promotor verfügt. Auf *B. licheniformis* DSM13 bezogen suggeriert dies, dass ComA und damit die Ausbildung der Kompetenz in *B. licheniformis* DSM13 anders kontrolliert wird als in *B. subtilis* (Auchtung *et al.*, 2006). Diese Annahme wird durch andere Untersuchungen am Kompetenzsystem von *B. licheniformis* DSM13 gestützt (Wollherr, 2010). Jakobs *et al.* zeigen außerdem, dass es einen direkten Zusammenhang zwischen der Ausbildung der genetischen Kompetenz und der Bildung und Ausscheidung von abbauenden Enzymen gibt (hier untersucht Glukanasen und Proteasen) (Jakobs *et al.*, 2014).

RapD zeigt neben einem möglichen SigA Promotor einen SigH Promotor welcher in geringem Maße Aktivität zeigt. Ein SigX Promotor, wie durch Huang und Helmann (Huang and Helmann, 1998) in *B. subtilis* beschrieben, konnte nicht gefunden werden. Ansonsten zeigt das *rapD/phrD* Paar das erwartete Verhalten, wo ab dem Übergang in die stationäre Phase, wahrscheinlich bedingt durch SigH, das Verhältnis von *phrD* zu *rapD* steigt. Auffällig ist die starke Aktivität von *phrD* im Vergleich zu den anderen *phr*-Genen. *RapG/phrG* zeigen das steigende Verhältnis von *phrG* zu *rapG* ebenfalls und besitzen die von Mcquade *et al.* und Jarmer *et al.* beschriebenen Promotoren. Das *rapI/phrI* Paar ist Teil eines konjugativen Transposons (ICE, *integrative and conjugative element*) (Lee *et al.*, 2012). Die Expression der Gene dieses Transposons wird durch *rapI* kontrolliert. Dieses Transposon wird von Lee *et al.* als wichtig für die Konjugation von Plasmiden ohne eigene Mobilisierungsmaschinerie beschrieben. Da *phrI* ein Repressor für *rapI* ist und *rapI* Aktivität für die Aktivierung des Transposons benötigt wird, müsste dies heißen dass mit Beginn der stationären Phase die Aktivität des Transposons abnimmt.

Die tatsächliche Aktivität der *rap*-Gene muss aber experimentell abgeklärt werden um verlässliche Aussagen über die Einflüsse dieser Gene auf das Verhalten des Organismus zu treffen.

10.5 Promotorvorhersagen

Die Promotorvorhersagen sind ein *proof of concept* für die Vorhersage von Transkriptionsstarts durch TraV wie auch ein Beleg dafür, dass die RNA-Seq basierte, genomweite Vorhersage von *loci* für Promotorbindestellen funktioniert. Die Untersuchung von Promotorbindestellen ist derzeitig immernoch begrenzt durch ihre schwierige Vorhersagbarkeit mit bioinformatischen Methoden und den hohen Kosten und dem Arbeitsaufwand der Labormethoden, wobei oftmals die geringe Menge an Labordaten die Erstellung von verlässlichen Modellen für die bioinformatischen Methoden erschwert. Mittels der RNA-Seq Methode können eine Vielzahl anö Beispielen für spezifische Organismen generiert werden. Der besondere Vorteil hierbei ist, dass diese Vorhersagen kein spezifisches Experiment benötigen, sondern standardmäßig aus einem normalen RNA-Seq

Experiment mit ausreichender *coverage* generiert werden können. Mittels reiner bioinformatischer Suche, basierend auf einem HMM, konnten Jarmer *et al.* (Jarmer *et al.*, 2001) in *B. subtilis* 2538 Kandidaten für SigA Bindestellen vorhersagen. Von diesen lagen 1127 Kandidaten innerhalb von 400 Basen *upstream* von Genen. Mittels Nimmersatt und TraV konnten in *B. licheniformis* DSM13 1317 Kandidaten für SigA Bindestellen gefunden werden, welche sich innerhalb von 50 Basen *upstream* von TSS Kandidaten befinden. Es konnten also vergleichbare Mengen an Vorhersagen getroffen werden, nur dass die Promotor Kandidaten dank der RNA-Seq Daten genauer lokalisiert werden konnten und sich *denovo* aus den TSS Kandidaten ergeben haben, also nicht abhängig von einem vorher kurierten Modell sind.

Der Nimmersatt Algorithmus ist ein erster Schritt um Promotor *patterns* basierend auf RNA-Seq Daten zu identifizieren. Eine denkbare Verbesserung wäre z.B. die automatische Generierung einer PWM für die vorhergesagten *patterns* und anschließende Anwendung dieser PWM auf die eigentlichen *seed* Sequenzen. Auf diese Weise sollten Sequenzen wieder in den Kandidatenpool zurückgeführt werden, wenn diese schlecht zur eigentlichen PWM passen, um so die Anzahl an Fehlzuordnungen zu reduzieren. Die Analyse der den *patterns* zugeordneten Proteine kann ebenfalls verbessert werden. Mit Operonvorhersagen kombiniert, könnten alle dem TSS unterstellten Proteine untersucht werden, anstatt nur des ersten Proteins nach dem TSS. Sequenziertechnologien, welche gesamte Transkripte am Stück sequenzieren können, wie z.B. PacBio, könnten so die Analyse von Regulons erheblich verbessern. Außerdem könnte eine aktuellere Alternative für COG, wie z.B. Gene Ontology (Harris *et al.*, 2004), die Klassizifierung der Proteine verbessern.

Neben den bekannten Promotor *patterns* konnten eine Vielzahl an *patterns* gefunden werden, die nicht zu σ-Faktorbindestellen passen. Viele der *patterns* basieren lediglich auf wenigen Sequenzen und könnten falsch positive Ergebnisse des MEME Algorithmus sein. Zusätzliche Heuristiken, wie z.B. bei σ-Faktoren der spezifische Abstand zwischen erkannten patterns, sind demnach nötig, um die Menge an Kandidaten zu reduzieren. Desweiteren sind Suchen denkbar, die außerhalb der σ-Faktor Bindestellen liegen. Diese UP-Elemente liegen im Bereich von -40 bis -90 Basen *upstream* vom TSS und können einen großen Einfluss auf die Transkriptionsstärke haben (Ross *et al.*, 1998).

Der Nimmersatt Ansatz zeigt die Möglichkeit auf, die den identifizierten *patterns* zugeordneten Gene in regulatorische Netzwerke einzugliedern. Untersuchungen mit *Saccharomyces cerevisiae* demonstrieren die Möglichkeiten eines solchen Ansatzes (Pilpel *et al.*, 2001), wo verschiedene *patterns* in Netzwerke eingeteilt werden konnten.

10.6 Prophagenaktivitätsbestimmung

Die Bestimmung von Prophagen in *B. licheniformis* DSM13 zeigt dass TraV nicht nur in RNA-Seq Experimenten, sondern allgemein in Experimenten, die NGS basierte *mappings* beinhalten, eingesetzt werden kann. Die Darstellung und die analytischen Methoden erlauben eine genaue Bestimmung der Prophagenbereiche sowie die Betrachtung der Aktivitätsveränderung der einzelnen Prophagenregionen in den Deletionsmutanten. Rein bioinformatische *Tools* zur Vorhersage von Prophagen wie PHAST (Zhou *et al.*, 2011) und Prophage Finder (Bose and Barber, 2006) konnten die Prophagenregionen ebenfalls identifizieren, waren jedoch nicht so genau in der Eingrenzung der Prophagenregionen wie eine manuelle Kuration. Außerdem können diese *tools* keine Betrachtung der Aktivität dieser Prophagen machen. Sie stellen aber eine gute Grundlage für die mit TraV und den experimentellen Daten mögliche, genauere Bestimmung dar.

Die Aktivität der BLi_Pp7 Prophagenregion ist in diesem Experiment nicht eindeutig untersuchbar. Dies begründet sich in der, für die verbesserte Transformierbarkeit, notwendigen Deletionen, die bei der Erstellung des MW3 Stamms durchgeführt wurden (Waschkau *et al.*, 2008). Bei diesen Deletionen wurden Teile der BLi_Pp7 Prophagenregion deletiert was dazu führen könnte, dass der Prophage inaktiv wird. Um die Aktivität des BLi_Pp7 Prophagen zu betrachten, wären DSM13-ΔBLi_Pp2 und eventuell DSM13-ΔBLi_Pp3 Mutanten notwendig.

10.7 Metatranskriptom einer Algenblüte aus der Nordsee

Metatranskriptomische Analysen stellen derzeit einen Grenzbereich der Möglichkeiten der RNA-Seq dar. In Experimenten haben Tarazona *et al.* (Tarazona *et al.*, 2011) gezeigt dass die Tiefe der Sequenzierung bei RNA-Seq Experimenten eine kritische Größe bei der Auswertung von Genaktivitäten darstellt. Bei Metatranskriptomen wird die Sequenzierleistung auf mehrere Organismen aufgeteilt, was die Sequenzierleistung pro Organismus reduziert. Dies ist gut erkennbar an der in dieser Untersuchung verwendeten Menge an *reads,* von denen ca. 2,3% bis 5,3% der sequenzierten *reads mapped* werden konnten. In den RNA-Seq Experimenten auf *B. licheniformis* DSM13 bewegt sich der prozentuale Anteil der *mapped reads* an der Gesamtsequenzierleistung zwischen ~5,7 bis 11,7% (Wiegand *et al.*, 2013) wobei diese nicht zusätzlich zwischen verschiedenen Organismen aufgeteilt werden.

Dennoch konnten ca. 94,6% des *P. temperata* RCA23 Genoms in diesem Experiment abgedeckt werden. Dies begründet sich in der Dominanz dieses Organismus in dem betrachteten Habitat (Giebel *et al.*, 2013). Cand. *P. ubique* HTCC1062 konnte zu 42,6%

abgedeckt werden. Interessanterweise ist HTCC2207 in der Algenblüte am Tag zu 89,1% abgedeckt während er in der Algenblüte in der Nacht nur zu 34,1% abgedeckt ist. Dies deutet auf die physiologischen Eigenarten der Organismen hin, welche nur unter bestimmten Bedingungen aktiv werden (Voget *et al.*, 2014).

Differentielle Expressionsanalysen sind Aufgrund der mangelnden Sequenziertiefe nicht aussagekräftig aufgrund der von Tarazona *et al.* beschriebenen Problematik. Dennoch sind Aussagen über die transkriptionelle Aktivität oder Inaktivität von Genen möglich, wie anhand der Photosynthesegene und den Stressproteinen gezeigt werden konnte.

Folglich sind Analysen von Metatranskriptomen in TraV möglich, jedoch ist die Aussagekraft aufgrund der derzeit möglichen Sequenziertiefen begrenzt. Voraussetzung für solche Analysen in TraV sind Referenzgenome für das *mapping* mit ausreichender Qualität wie Beispielsweise *P. temperata* RCA23. Dieser Ansatz funktioniert bei Metatranskriptomen, wo solche qualitativ hochwertigen Referenzgenome vorliegen. Sollten keine solchen Referenzgenome vorliegen, kann TraV derzeitig nicht verwendet werden. Dies liegt an der in 10.2 beschriebenen Problematik mit ungeschlossenen Genomen.

11 Zusammenfassung

- Das TraV *tool* bietet eine speichereffiziente und performante Analysesoftware für die Auswertung von RNA-Seq Experimenten. Der Fokus liegt auf die Entdeckung von bisher nicht annotierten regulatorischen *features* und Transkriptionsstartpunkten (TSS). Dieser Fokus und die Fähigkeit viele Datensätze in den Analysen zu kombinieren macht es zu einer guten Ergänzung zu bereits bestehenden *tools* zur RNA-Seq Auswertung. Die Fähigkeit der RNA-Seq, die Reaktionen eines Organismus auf Stimuli aufzuzeigen liefert Ansatzpunkte für weiterführende Experimente. Visualisierungs und Analysetools wie TraV geben durch die Auf- und Bearbeitung der großen Datenmengen von RNA-Seq Experimenten entscheidende Hilfestellung bei der Auswertung dieser Daten.

- Die TraV Analysen zeigen die Vorhersagekraft der Kombination von bioinformatischen und laborbiologischen Methoden. Diese erlaubt Einblicke in die Physiologie, die ohne diese Kombination nur schwer oder nicht möglich sind: i) Vorhersagen von regulatorischen RNAs sowie die Beschreibung von deren Einfluss auf die Gene unter ihrer Kontrolle, ii) Identifikation von differentiell exprimierten Genen und die Verbindung dieser differentiellen Expression mit bekannten Regulatoren und den Wachstumsbedingungen und Wachstumsphase, iii) das Auffinden von Promotorbindestellen basierend auf den Expressionsprofilen des Organismus sowie iv) die Aufklärung von möglichen *multilayer* Regulationen in Verbindung mit den Wachstumsbedingungen und Wachstumsphasen.

- Der Nimmersatt Ansatz zeigt die Möglichkeit, basierend auf den TraV Vorhersagen bestehende *patternfinding tools* (in diesem Fall MEME) anhand experimenteller Daten zu dirigieren. Dies verbessert die Vorhersagekraft dieser *tools*, indem kurierte Kandidaten als *input* bereitgestellt werden. Nimmersatt liefert durch die COG Analyse der *pattern* assoziierten Proteine Indizien für die Rekonstruktion regulatorischer Netzwerke.

- TraV ist neben der Transkriptomsequenzierung in weiteren, verwandten Gebieten wie der Metatranskriptomik und neuen Gebieten wie der Prophagenaktivititäts-bestimmung erfolgreich eingesetzt worden. Damit wurde die Nützlichkeit vielseitiger Visualisierungs- und Vorhersagetools für NGS basierte Daten in Korrelation zu genomisch kodierten biologischen Features gezeigt.

12 Literaturverzeichnis

Aird,D. *et al.* (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.*, **12**, R18.

Altschul SF, Gish W, Miller W, Myers EW,L.D. (1990) Basic local alignment search tool. *J Mol Biol.*, **215**, 403–410.

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R106.

Ansong,C. *et al.* (2013) A multi-omic systems approach to elucidating Yersinia virulence mechanisms. *Mol. Biosyst.*, **9**, 44–54.

Auchtung,J.M. *et al.* (2006) Modulation of the ComA-dependent quorum response in Bacillus subtilis by multiple Rap proteins and Phr peptides. *J. Bacteriol.*, **188**, 5273–85.

Auger,S. *et al.* (2002) The metIC operon involved in methionine biosynthesis in Bacillus subtilis is controlled by transcription antitermination. *Microbiology*, **148**, 507–18.

Backofen,R. and Hess,W.R. (2010) Computational prediction of sRNAs and their targets in bacteria. *RNA Biol.*, **7**, 33–42.

Bailey,T.L. *et al.* (2006) MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.*, **34**, W369–73.

Bailey,T.L. (1995) Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization. *Mach. Learn.*, **21**, 51–80.

Belasco,J.G. and Higgins,C.F. (1988) Mechanisms of mRNA decay in bacteria: a perspective. *Gene*, **72**, 15–23.

Bolger,A.M. *et al.* (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 1–7.

Bose,M. and Barber,R.D. (2006) Prophage Finder: a prophage loci prediction tool for prokaryotic genome sequences. *In Silico Biol.*, **6**, 223–7.

Bouvier,M. *et al.* (2008) Small RNA binding to 5’ mRNA coding region inhibits translational initiation. *Mol. Cell*, **32**, 827–37.

Burge,S.W. *et al.* (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.*, **41**, D226–32.

Busby,S. and Ebright,R.H. (1994) Promoter structure, promoter recognition, and transcription activation in prokaryotes. *Cell*, **79**, 743–6.

Campbell,E. a *et al.* (2002) Structure of the bacterial RNA polymerase promoter specificity sigma subunit. *Mol. Cell*, **9**, 527–39.

Cao,M. *et al.* (2002) Defining the Bacillus subtilis sigma(W) regulon: a comparative analysis of promoter consensus search, run-off transcription/macroarray analysis (ROMA), and transcriptional profiling approaches. *J. Mol. Biol.*, **316**, 443–57.

Carver,T. *et al.* (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics*, **28**, 464–9.

Carver,T. *et al.* (2009) DNAPlotter: circular and linear interactive genome visualization. *Bioinformatics*, **25**, 119–20.

Casjens,S. (2003) Prophages and bacterial genomics: what have we learned so far? *Mol. Microbiol.*, **49**, 277–300.

Cho,J. and Giovannoni,S.J. (2004) Cultivation and Growth Characteristics of a Diverse Group of Oligotrophic Marine Gammaproteobacteria. *Appl. Environ. Microbiol.*, **70**, 432–440.

Chowdhury,S. *et al.* (2003) Temperature-controlled structural alterations of an RNA thermometer. *J. Biol. Chem.*, **278**, 47915–21.

Crooks,G.E. *et al.* (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–90.

Darty,K. *et al.* (2009) VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.

Das,M.K. and Dai,H.-K. (2007) A survey of DNA motif finding algorithms. *BMC Bioinformatics*, **8 Suppl 7**, S21.

Davis,B.M. and Waldor,M.K. (2007) RNase E-dependent processing stabilizes MicX, a Vibrio cholerae sRNA. *Mol. Microbiol.*, **65**, 373–85.

Denschlag,C. (2010) Untersuchung zur RNA-basierten Regulation in Bacillus licheniformis.

Deutscher,M.P. (1988) The metabolic role of RNases. *TIBS*, **13**, 136–139.

Dietrich,S. (2009) Untersuchung zur Biologie von RNA basierten Regulatoren in Bacilli.

Van Dijk,E.L. *et al.* (2014) Ten years of next-generation sequencing technology. *Trends Genet.*, **30**.

Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–5.

Eddy,S.R. *et al.* (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res. Vol. 22. No.11 2079-2088*, **22**, 2079–2088.

Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–88.

Eiamphungporn,W. and Helmann,J.D. (2008) The Bacillus subtilis sigma(M) regulon and its contribution to cell envelope stress responses. *Mol. Microbiol.*, **67**, 830–48.

Eichenberger,P. *et al.* (2004) The program of gene transcription for a single differentiating cell type during sporulation in Bacillus subtilis. *PLoS Biol.*, **2**, e328.

Eichenberger,P. *et al.* (2003) The σE Regulon and the Identification of Additional Sporulation Genes in Bacillus subtilis. *J. Mol. Biol.*, **327**, 945–972.

Ellinger,T. *et al.* (1994) Stalling of Escherichia coli RNA Polymerase in the +6 to +12 Region in Vivo is Associated with Tight Binding to Consensus Promoter Elements. *J. Mol. Biol.*, **239**, 455–465.

Ellington, Andrew D; Szostak,J.W. (1990) In vitro selection of RNA molecules that bind specific ligands. *Nature*, **346**, 818–822.

Feklistov,A. and Darst,S.A. (2011) Structural basis for promoter-10 element recognition by the bacterial RNA polymerase σ subunit. *Cell*, **147**, 1257–69.

Frohman,M.A. *et al.* (1988) Rapid production of full-length cDNAs from rare transcripts: Amplification using single gene-specific oligonucleotide primer. *Proc Natl Acad Sci USA*, **85**, 8998–9002.

Giebel,H.-A. *et al.* (2013) Planktomarina temperata gen. nov., sp. nov., belonging to the globally distributed RCA cluster of the marine Roseobacter clade, isolated from the German Wadden Sea. *Int. J. Syst. Evol. Microbiol.*, **63**, 4207–17.

Giovannoni,S.J. *et al.* (2005) Genome streamlining in a cosmopolitan oceanic bacterium. *Science*, **309**, 1242–5.

Göpel,Y. *et al.* (2013) Targeted decay of a regulatory small RNA by an adaptor protein for RNase E and counteraction by an anti-adaptor RNA. *Genes Dev.*, **27**, 552–64.

Gruber,A.R. *et al.* (2010) RNAz 2.0: improved noncoding RNA detection. *Pac. Symp. Biocomput.*, 69–79.

Grundy,F.J. and Henkin,T.M. (1998) The S box regulon: a new global transcription termination control system for methionine and cysteine biosynthesis genes in Gram-positive bacteria. *Mol. Microbiol.*, **30**, 737–749.

Hardcastle,T.J. and Kelly,K.A. (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, **11**, 422.

Harris,M. a *et al.* (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.*, **32**, D258–61.

Hazra,A.B. *et al.* (2011) A missing enzyme in thiamin thiazole biosynthesis: identification of TenI as a thiazole tautomerase. *J. Am. Chem. Soc.*, **133**, 9311–9.

Hindley,J. (1967) Fractionation of 32P-labelled Ribonucleic Acids on Polyacryl- amide Gels and their Characterization by Fingerprinting. *J. Mol. Biol.*, **30**, 125–136.

Huang,X. and Helmann,J.D. (1998) Identification of target promoters for the Bacillus subtilis sigma X factor using a consensus-directed search. *J. Mol. Biol.*, **279**, 165–73.

Jahn,N. *et al.* (2012) BsrG/SR4 from Bacillus subtilis- the first temperature-dependent type I toxin-antitoxin system. *Mol. Microbiol.*, **83**, 579–98.

Jahn,N. and Brantl,S. (2013) One antitoxin--two functions: SR4 controls toxin mRNA decay and translation. *Nucleic Acids Res.*, **41**, 9870–80.

Jakobs,M. *et al.* (2014) Unravelling the genetic basis for competence development of auxotrophic Bacillus licheniformis 9945A strains. *Microbiology*, **160**, 2136–47.

Jarmer,H. *et al.* (2001) Sigma A recognition sites in the Bacillus subtilis genome. *Microbiology*, **147**, 2417–24.

Kent,W.J. *et al.* (2010) BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*, **26**, 2204–7.

Kent,W.J. *et al.* (2002) The Human Genome Browser at UCSC. *Genome Res.*, **12**, 996–1006.

Kingsford,C.L. *et al.* (2007) Rapid, accurate, computational discovery of Rho-independent transcription terminators illuminates their relationship to DNA uptake. *Genome Biol.*, **8**, R22.

Krzywinski,M. *et al.* (2009) Circos: an information aesthetic for comparative genomics. *Genome Res.*, **19**, 1639–45.

Kumar A, Malloch RA, Fujita N, Smillie DA, Ishihama A,H.R. (1993) The minus 35-recognition region of Escherichia coli sigma 70 is inessential for initiation of transcription at an “extended minus 10” promoter. *J Mol Biol.*, **232**, 406–18.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–9.

Lee,C. a *et al.* (2012) The Bacillus subtilis conjugative transposon ICEBs1 mobilizes plasmids lacking dedicated mobilization functions. *J. Bacteriol.*, **194**, 3165–72.

Lehnik-Habrink,M. *et al.* (2012) RNA degradation in Bacillus subtilis: an interplay of essential endo- and exoribonucleases. *Mol. Microbiol.*, **84**, 1005–17.

Levitsky,V.G. *et al.* (2007) Effective transcription factor binding site prediction using a combination of optimization, a genetic algorithm and discriminant analysis to capture distant interactions. *BMC Bioinformatics*, **8**, 481.

Lewin,B. (2008) Genes IX.

Li,B. *et al.* (2010) RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, **26**, 493–500.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–9.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–60.

Li,L. *et al.* (2007) GAPWM: a genetic algorithm method for optimizing a position weight matrix. *Bioinformatics*, **23**, 1188–94.

Linder,P. *et al.* (2014) Transcriptome-wide analyses of 5'-ends in RNase J mutants of a gram-positive pathogen reveal a role in RNA maturation, regulation and degradation. *PLoS Genet.*, **10**, e1004207.

Lindgreen,S. *et al.* (2006) Measuring covariation in RNA alignments: physical realism improves information measures. *Bioinformatics*, **22**, 2988–95.

Liu,Y. *et al.* (2013) RNA-seq differential expression studies: more sequence or more replication? *Bioinformatics*, **30**, 301–304.

MacLellan,S.R. *et al.* (2008) A previously unidentified sigma factor and two accessory proteins regulate oxalate decarboxylase expression in Bacillus subtilis. *Mol. Microbiol.*, **69**, 954–67.

Mansilla,M.C. *et al.* (2000) Transcriptional Control of the Sulfur-Regulated cysH Operon , Containing Genes Involved inl-Cysteine Biosynthesis in Bacillus subtilis. *J. Bacteriol.*, **182**, 5885–5892.

Marco-sola,S. *et al.* (2012) The GEM mapper : fast , accurate and versatile alignment by filtration. *Nat. Methods*, **9**.

Mascher,T. *et al.* (2007) Regulatory overlap and functional redundancy among Bacillus subtilis extracytoplasmic function sigma factors. *J. Bacteriol.*, **189**, 6919–27.

Mattick,J.S. (2004) RNA regulation : a new genetics ? *Nat. Rev. Genet.*, **5**, 316–323.

Mccue,L.A. *et al.* (2001) Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res.*, **29**, 774–782.

McKenney,P.T. *et al.* (2013) The Bacillus subtilis endospore: assembly and functions of the multilayered coat. *Nat. Rev. Microbiol.*, **11**, 33–44.

Mcquade,R.S. *et al.* (2001) Control of a Family of Phosphatase Regulatory Genes (phr) by the Alternate Sigma Factor Sigma-H of Bacillus subtilis. *J. Bacteriol.*, **183**, 4905–4909.

Merrick,M.J. (1993) In a class of its own — the RNA polymerase sigma factor sigma54. *Mol. Microbiol.*, **10**, 903–909.

Michna,R.H. *et al.* (2014) SubtiWiki-a database for the model organism Bacillus subtilis that links pathway, interaction and expression information. *Nucleic Acids Res.*, **42**, D692–8.

Miranda-Ríos,J. *et al.* (2001) A conserved RNA structure (thi box) is involved in regulation of thiamin biosynthetic gene expression in bacteria. *Proc. Natl. Acad. Sci. U. S. A.*, **98**, 9736–41.

Mirel,D.B. and Chamberlin,M.J. (1989) The Bacillus subtilis flagellin gene (hag) is transcribed by the sigma 28 form of RNA Polymerase. *J. Bacteriol.*, **171**, 3095–3101.

Mizuno,T. *et al.* (1984) A unique mechanism regulating gene expression: translational inhibition by a complementary RNA transcript (micRNA). *Proc. Natl. Acad. Sci. U. S. A.*, **81**, 1966–70.

Moran,M.A. *et al.* (2013) Sizing up metatranscriptomics. *ISME J.*, **7**, 237–43.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, 1–8.

Munch,R. (2003) PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res.*, **31**, 266–269.

Nakano,T. *et al.* (2014) Plausible Novel Ribose Metabolism Catalyzed by Enzymes of the Methionine Salvage Pathway in Bacillus subtilis. *Biosci. Biotechnol. Biochem.*, **77**, 1104–1107.

Narberhaus,F. (2009) MicroMeeting Report. *Mol. Microbiol.*, 1–9.

Nawrocki,E.P. *et al.* (2009) Infernal 1.0: inference of RNA alignments. *Bioinformatics*, **25**, 1335–7.

Nicolas,P. *et al.* (2012) Condition-Dependent Transcriptome Reveals High-Level Regulatory Architecture in Bacillus subtilis. *Science*, **335**, 1103–1106.

Niedringhaus,T.P. *et al.* (2011) Landscape of next-generation sequencing technologies. *Anal. Chem.*, **83**, 4327–41.

Nudler,E. and Mironov,A.S. (2004) The riboswitch control of bacterial metabolism. *Trends Biochem. Sci.*, **29**, 11–7.

Overbeek,R. (2003) The ERGOTM genome analysis and discovery system. *Nucleic Acids Res.*, **31**, 164–171.

Paget,M.S.B. and Helmann,J.D. (2003) The 70 family of sigma factors. *Genome Biol.*, **4**, 1–6.

Pang,K.C. *et al.* (2006) Rapid evolution of noncoding RNAs: lack of conservation does not mean lack of function. *Trends Genet.*, **22**, 1–5.

Paprotka,T. *et al.* (2012) Third Generation Sequencer: Neue Möglichkeiten für die de novo-Assemblierung. *BIOspektrum*, **18**, 524–526.

Pedersen,A.G. *et al.* (1999) The biology of eukaryotic promoter prediction--a review. *Comput. Chem.*, **23**, 191–207.

Perego,M. (2013) Forty years in the making: understanding the molecular mechanism of peptide regulation in bacterial development. *PLoS Biol.*, **11**, 1–5.

Perego,M. and Hoch,J. a (1996) Cell-cell communication regulates the effects of protein aspartate phosphatases on the phosphorelay controlling development in Bacillus subtilis. *Proc. Natl. Acad. Sci. U. S. A.*, **93**, 1549–53.

Peterson,E.S. *et al.* (2012) VESPA: software to facilitate genomic annotation of prokaryotic organisms through integration of proteomic and transcriptomic data. *BMC Genomics*, **13**, 131.

Pilpel,Y. *et al.* (2001) Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat. Genet.*, **29**, 153–9.

Podkaminski,D. and Vogel,J. (2010) Small RNAs promote mRNA stability to activate the synthesis of virulence factors. *Mol. Microbiol.*, **78**, 1327–31.

Predich,M. *et al.* (1992) Bacillus subtilis early sporulation genes kinA, spo0F, and spo0A are transcribed by the RNA polymerase containing sigma H. *J. Bacteriol.*, **174**, 2771–8.

Quevillon,E. *et al.* (2005) InterProScan: protein domains identifier. *Nucleic Acids Res.*, **33**, W116–20.

Ross,W. *et al.* (1998) Escherichia coli Promoters with UP Elements of Different Strengths : Modular Structure of Bacterial Promoters. *J. Bacteriol.*, **180**, 5375–5383.

Ruffalo,M. *et al.* (2011) Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics*, **27**, 2790–6.

Schallmey,M. *et al.* (2004) Developments in the use of Bacillus species for industrial production. *Can. J. Microbiol.*, **50**, 1–17.

Sekowska,A. and Danchin,A. (2002) The methionine salvage pathway in Bacillus subtilis. *BMC Microbiol.*, **2**, 8.

Shahbabian,K. *et al.* (2009) RNase Y, a novel endoribonuclease, initiates riboswitch turnover in Bacillus subtilis. *EMBO J.*, **28**, 3523–33.

Sharma,C.M. *et al.* (2010) The primary transcriptome of the major human pathogen Helicobacter pylori. *Nature*, **464**, 250–5.

Shingaki,R. *et al.* (2003) Chromosome DNA fragmentation and excretion caused by defective prophage gene expression in the early-exponential-phase culture of Bacillus subtilis. *Can. J. Microbiol.*, **49**, 313–325.

Silvaggi,J.M. *et al.* (2006) Genes for Small , Noncoding RNAs under Sporulation Control in Bacillus subtilis. *J. Bacteriol.*, **188**, 532–541.

Sonenshein,A.L. *et al.* (2002) Bacillus subtilis and Its Closest Relatives ASM Press, Washington, DC.

Staroń,A. *et al.* (2009) The third pillar of bacterial signal transduction: classification of the extracytoplasmic function (ECF) sigma factor protein family. *Mol. Microbiol.*, **74**, 557–81.

Steuten,B. *et al.* (2013) 6S RNA: recent answers - future questions. *Mol. Microbiol.*

Storz,G. *et al.* (2004) Controlling mRNA stability and translation with small, noncoding RNAs. *Curr. Opin. Microbiol.*, **7**, 140–4.

Tarazona,S. *et al.* (2011) Differential expression in RNA-seq: a matter of depth. *Genome Res.*, **21**, 2213–23.

Tatusov,R.L. *et al.* (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res.*, **29**, 22–8.

Thorstenson,Y.R. *et al.* (1998) An Automated Hydrodynamic Process for Controlled , Unbiased DNA Shearing. *Genome Res.*, **8**, 848–855.

Thorvaldsdóttir,H. *et al.* (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform.*, **14**, 178–92.

Thürmer,A. (2014) Next Generation Sequencing in der mikrobiellen (Meta) Genomforschung. *BIOspektrum*, 168–171.

Tjaden,B. (2008) Prediction of small, noncoding RNAs in bacteria using heterogeneous data. *J. Math. Biol.*, **56**, 183–200.

Toms,A. V *et al.* (2005) Structural characterization of the regulatory proteins TenA and TenI from Bacillus subtilis and identification of TenA as a thiaminase II. *Biochemistry*, **44**, 2319–29.

Tomsic,J. *et al.* (2008) Natural variability in S-adenosylmethionine (SAM)-dependent riboswitches: S-box elements in bacillus subtilis exhibit differential sensitivity to SAM In vivo and in vitro. *J. Bacteriol.*, **190**, 823–33.

Trapnell,C. *et al.* (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**, 511–5.

Trotochaud,A.E. and Wassarman,K.M. (2005) A highly conserved 6S RNA structure is required for regulation of transcription. *Nat. Struct. Mol. Biol.*, **12**, 313–9.

Tucker,B.J. and Breaker,R.R. (2005) Riboswitches as versatile gene control elements. *Curr. Opin. Struct. Biol.*, **15**, 342–8.

Tveit,A. *et al.* (2014) Metatranscriptomic analysis of Arctic peat soil microbiota. *Appl. Environ. Microbiol.*, DOI: 10.1128/AEM.01030–14.

Veith,B. *et al.* (2004) The complete genome sequence of Bacillus licheniformis DSM13, an organism with great industrial potential. *J. Mol. Microbiol. Biotechnol.*, **7**, 204–11.

Viegas,S.C. *et al.* (2007) Characterization of the role of ribonucleases in Salmonella small RNA decay. *Nucleic Acids Res.*, **35**, 7651–64.

Vitreschak,A.G. *et al.* (2002) Regulation of riboflavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Res.*, **30**, 3141–3151.

Vockenhuber,M.-P. *et al.* (2011) Deep sequencing-based identification of small non-coding RNAs in Streptomyces coelicolor. *RNA Biol.*, **8**, 468–477.

Voget,S. *et al.* (2014) Adaptation of an abundant Roseobacter RCA organism to pelagic systems revealed by genomic and transcriptomic analyses. *ISME J.*, DOI: 10.1038/ismej.2014.134.

Wagner,G.P. *et al.* (2012) Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci.*, **131**, 281–5.

Wang,Z. *et al.* (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.

Waschkau,B. *et al.* (2008) Generation of readily transformable Bacillus licheniformis mutants. *Appl. Microbiol. Biotechnol.*, **78**, 181–8.

Washietl,S. *et al.* (2005) Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. U. S. A.*, **102**, 2454–9.

Wemheuer,B. *et al.* (2014) Impact of a phytoplankton bloom on the diversity of the active bacterial community in the southern North Sea as revealed by metatranscriptomic approaches. *FEMS Microbiol. Ecol.*, **87**, 378–89.

Wiegand,S. *et al.* (2013) RNA-Seq of Bacillus licheniformis: active regulatory RNA features expressed within a productive fermentation. *BMC Genomics*, **14**, 667.

Wiegeshoff,F. *et al.* (2006) Sigma L Is Important for Cold Shock Adaptation of Bacillus subtilis. *J. Bacteriol.*, **188**, 3130–3133.

Winkler,W. *et al.* (2002) Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. *Nature*, **419**, 952–6.

Winkler,W.C. and Breaker,R.R. (2005) Regulation of bacterial gene expression by riboswitches. *Annu. Rev. Microbiol.*, **59**, 487–517.

Wollherr,A. (2010) Komparative Genomanalyse zur Stammoptimierung produktionsnaher Bacillus-Stämme.

Wurtzel,O. *et al.* (2010) A single-base resolution map of an archaeal transcriptome. *Genome Res.*, **20**, 133–41.

Yoshida,K.I. *et al.* (1997) Organization and transcription of the myo-inositol operon , iol , of Bacillus subtilis. *J. Bacteriol.*, **179**, 4591–4598.

Zhang,Y. *et al.* (2004) Conservation analysis of small RNA genes in Escherichia coli. *Bioinformatics*, **20**, 599–603.

Zhou,Y. *et al.* (2011) PHAST: a fast phage search tool. *Nucleic Acids Res.*, **39**, 347–52.

Ziady,A.G. and Kinter,M. (2009) Protein sequencing with tandem mass spectrometry. *Methods Mol. Biol.*, **544**, 325–41.

Zuber,U. *et al.* (2001) Putative Sigma Factor SigI (YkoZ) of Bacillus subtilis Is Induced by Heat Shock. *J. Bacteriol.*, **183**, 1472–1475.

13 Publikationen mit Beiträgen aus dieser Dissertation

Dietrich,S. *et al.* (2014) TraV: A Genome Context Sensitive Transcriptome Browser. *PLoS One*, **9**, DOI: 10.1371/journal.pone.0093677.

Wiegand,S. *et al.* (2013) RNA-Seq of *Bacillus licheniformis*: active regulatory RNA features expressed within a productive fermentation. *BMC Genomics*, **14**, 667.

Voget,S. *et al.* (2014) Adaptation of an abundant *Roseobacter* RCA organism to pelagic systems revealed by genomic and transcriptomic analyses. *ISME J.*, DOI: 10.1038/ismej.2014.134.

Hertel et al, submitted 2014, Genome-based identification of active prophage regions by next generation sequencing in *Bacillus licheniformis* DSM13

Lebenslauf

Persönliche Daten

Name	Sascha Dietrich
Geburtsdatum	10.07.1984
Geburtsort	Hildesheim
Staatsangehörigkeit	Deutsch
Familienstand	Ledig

Wissenschaftlicher Werdegang

Jul 2010 – Jan 2015	**Dissertation** in der Gruppe von Dr. Heiko Liesegang mit dem Titel "Analyse und Charakterisierung regulatorischer Vorgänge in *Bacillus licheniformis*"
Feb 2009 – Nov 2009	**Diplomarbeit** angefertigt in der Gruppe von Dr. Heiko Liesegang mit dem Titel "Untersuchung zur Biologie von RNA basierten Regulatoren in *Bacilli*"
Sep 2006 – Nov 2009	**Hauptstudium** Biologie an der Georg August Universität Göttingen, mit dem Hauptfach Mikrobiologie und den Nebenfächern Bioinformatik und Pathologie
Okt 2004 – Sep 2006	**Vordiplom** in Biologie an der Georg August Universität Göttingen
Aug 1997 – Mai 2004	**Allgemeine Hochschulreife** am Scharnhorstgymnasium Hildesheim
Aug 1995 – Jun 1997	**Schüler** Orientierungsstufe Ost Hildesheim
Aug 1991 – Jun 1995	**Schüler** Grundschule Holle