



universität
wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

Unsupervised construction, evaluation and visualisation of
RNA family models.

verfasst von / submitted by

Mag. rer. nat. Florian Eggenhofer

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (PhD)

Wien, 2016 / Vienna, 2016

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on the student
record sheet:

A 794 685 490

Dissertationsgebiet lt. Studienblatt /
field of study as it appears on the student record sheet:

Molekulare Biologie

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Phys. Dr. Ivo L. Hofacker

Acknowledgments/Danksagung

I would like to acknowledge some outstanding persons and institutions that supported me and my work for this thesis.

Ivo, my sincere thanks for sharing your wisdom and knowledge, but also for your good-will and patience. Your enthusiasm for science has set an shining example for me, what it means to be a scientist. It has been a privilege to have you as an advisor, for which I will be always grateful.

Christian, I want to express my deepest gratitude for the countless instances where you have supported me. Your expertise and dedication have been truly inspiring for me. A heartily thank you for introducing me to functional programming and *Haskell*. I consider myself very fortunate having you as co-advisor and collaborator.

I want to thank my collaborators from the **ViennaNGS** team:

Michael for initiating and leading the **ViennaNGS** project, for guiding me through the intricacies hub construction (and for *Speck*)

Jörg, for hacking, thinking and discussing with me, instantly finding the most elusive of bugs and being the truest of friends.

Fabian, for sharing my fascination with small *RNAs*, discussing with me and being *symbadisch*.

I want to thank the members of my PhD-committee, **Peter F. Stadler** and **Renée Schroeder** for their advise, encouragement and suggestions.

I want to express my deepest gratitude for the administrative support from:

Judith, for being the kind and compassionate soul of the institute and the excellent way she manages the administration.

Richard, for being a real-life Scotty and for supporting me with technical and personal advise over the years. Hack on!

Gerlinde, for her support in the PhD program and overcoming of bureau-

cratic odds.

I want to thank **Sita Saunders** for her time and help in improving the text quality of the foreword and the discussion.

Furthermore I want to thank:

The **reviewers** of the publications included in this thesis, for their contributions to the manuscripts and the tools.

The **reviewers** of this thesis for consenting to review this manuscript and for their corresponding investment of effort and time.

The three awesome "Musketees", **Stefan, Peter** and **Jörg** (again) for sharing the PhD experience together with me since the PhD selection.

The crowd of friendly people, who were not directly involved in my thesis, at the **TBI**. I really appreciated the time with you.

Rolf for his generosity and for giving me a new scientific home at the **University of Freiburg**.

I also want to thank the funding agencies **DFG - Deutsche Forschungsgemeinschaft**, **SNF - Schweizer Nationalfond** and **FWF - Der Wissenschaftsfond** and the **University of Vienna** for funding.

Finally my **Mum** and **Dad** for their unwavering support and believing in me.

Abstract

RNA performs important functions in all organisms, for example mediating gene expression. *RNAs* are often evolutionary conserved over large set of species, giving rise to families of homologous *RNA* genes. These *RNA* families exhibit not only sequence similarity, but are often characterized by strong conservation of the *RNA* structure.

Computationally, *RNA* families are represented by *RNA*-family models, also known as covariance models. Covariance models capture structure and sequence of the family in a probabilistic model. They enable the prediction of additional, previously unknown, members of the *RNA*-family from genomic sequences. This allows a knowledge transfer between organisms and helps in designing experiments.

Up to now *RNA*-family models were constructed by manual collection and curation, or automatic solutions for a few specific *RNA* families. The peer-reviewed publication for "**RNAlien** - Unsupervised *RNA*-family model construction" introduces a novel method to automatically construct such models, in principle for any *RNA* sequence. **RNAlien**, starting from a single input sequence collects potential family member sequences by multiple iterations of homology search. *RNA*-family models are fully automatically constructed for the found sequences.

The quality of *RNA*-family models and their performance in homology search depends on several factors. **RNAlien** evaluates both the models as well as the aligned sequences used to build them, to provide as much information about the model as possible. However this takes only the novel model itself into consideration, but does not investigate it in context with other models.

The following manuscript, with the title "**CMCompare** webserver: comparing *RNA* families via covariance models", addresses the comparison between models. This allows to identify models with poor specificity and to explore the relationship between models. Visualisation of family relationships helps in identifying candidates for clans, groups of biologically related families.

Moreover the thesis presents a novel tool to visualise and compare the taxonomy of of found *RNA*-family members, called **TaxonomyTools**.

Family member sequences found by **RNAlien** during the model construction process are also a useful starting point for investigating families. **UCSC genome browser** hubs visualise the found family members in their genetic context, showing traits like orthology. Methods to constructs such hubs were contributed to the publication "**ViennaNGS: A toolbox for building efficient next-generation sequencing analysis pipelines**" and are also presented in the thesis.

Zusammenfassung

RNA-Familien werden in den Computerwissenschaften durch *RNA*-Familien Modelle, auch bekannt als Kovarianz-Modelle repräsentiert. Kovarianz-Modelle bilden Struktur und Sequenz der Familie als statistisches Modell ab. Sie machen es möglich weitere, zuvor unbekannte, Vertreter der *RNA* Familie in genomischen Sequenzen zu identifizieren. Dieser Vorgang ermöglicht es bekanntes Wissen und experimentelle Ergebnisse von einem auf den anderen Organismus zu transferieren und vereinfacht das Design neuer Experimente.

In der Vergangenheit wurden *RNA*-Familien Modelle durch manuelles Sammeln und Verfeinern, oder durch automatische Lösungen für einige wenige spezielle *RNA* Familien konstruiert. Die Publikation "**RNAlien** - Unsupervised *RNA*-family model construction" stellt eine neue Methode zum automatischen Konstruieren solcher Modelle, prinzipiell für jede *RNA* Sequenz, vor. **RNAlien**, ausgehend von einer einzelnen Eingabesequenz, sammelt potentielle Familienmitglieder durch multiple Iteration von Homologiesuche. *RNA*-Familien Modelle werden automatisch für die gefundenen Sequenzen gebaut.

Die Qualität von *RNA*-Familien Modellen und ihre Leistungsfähigkeit in der Homologiesuche hängt von verschiedenen Faktoren ab. **RNAlien** wertet sowohl die Modelle, als auch die alignierten Sequenzen die zum Bau der Modelle verwendet wurden, aus um so viel Information wie möglich zur Verfügung zu stellen. Dies berücksichtigt allerdings nur das neukonstruierte Modell und setzt es nicht in Beziehung zu anderen Modellen.

Die folgende Publikation, mit dem Titel "**CMCompare** webserver: comparing *RNA* families via covariance models", behandelt den Vergleich zwischen Modellen. Dies erlaubt die Identifizierung von Modellen mit schlechter Spezifität und die Untersuchung von Beziehungen zwischen Modellen. Visualisierung dieser Zusammenhänge hilft bei der Identifizierung von Kandidaten für Clans, Gruppen biologisch verknüpfter Familien.

Darüberhinaus wird ein Programmpaket, mit dem Namen **TaxonomyTools**, vorgestellt, welches die Visualisierung und den Vergleich der Taxonomie von gefundenen *RNA* Familien Mitgliedern ermöglicht.

Sequenzen von Familienmitglieder, die von **RNAlien** während des Konstruk-

tionsprozesses identifiziert wurden, sind ein Ausgangspunkt für die weitere Untersuchung der Familie. **UCSC genome browser** hubs visualisieren die gefundenen Familienmitglieder in ihrem genomischen Kontext, was Eigenschaften wie zum Beispiel Orthologie sichtbar macht. Methoden um solche Hubs zu bauen wurden als Beitrag mit der Publikation "**ViennaNGS: A toolbox for building efficient next-generation sequencing analysis pipelines**" veröffentlicht und werden hier präsentiert.

Contents

List of Figures	XI
------------------------	-----------

List of Tables	XIII
-----------------------	-------------

1 Foreword	1
-------------------	----------

2 Theoretical Background	3
---------------------------------	----------

2.1 <i>RNA</i> biology	5
2.1.1 Sequence	5
2.1.2 Secondary structure	6
2.1.3 Tertiary structure	9
2.1.4 Quaternary structure	11
2.1.5 RNA function	12
2.1.6 Homology	13
2.1.7 Phylogenetics	14
2.1.8 Taxonomy	16
2.1.9 <i>RNA</i> groups	18
2.2 Sequence alignment	21
2.2.1 Pairwise sequence alignment	21
2.2.2 Multiple sequence alignment	26
2.3 Probabilistic models	30
2.3.1 Hidden Markov models	30
2.3.2 Stochastic Context Free Grammar	34
2.4 <i>RNA</i> -family models	36
2.4.1 Infernal	36
2.4.2 Rfam - <i>RNA</i> -family database	39
2.5 Homology search	42
2.5.1 BLAST	43
2.5.2 Expected value	44
2.5.3 nhmmer	45
2.5.4 cmsearch	45

3	RNA-family model construction	46
3.1	Construction of <i>RNA</i> families and Clans	47
3.1.1	Seed alignment	47
3.1.2	Consensus secondary structure	47
3.1.3	Alignment to model	48
3.1.4	<i>RNA</i> -family clans	48
3.1.5	Full alignment	49
3.2	Conservation of <i>Rfam</i> families	50
4	RNA family model evaluation	54
4.1	CMCompare	54
5	RNA-family member visualization	56
5.1	Genome browser visualization	57
5.1.1	Trackhub construction	58
5.1.2	Assembly hub construction	60
5.2	<i>RNA</i> -family members and taxonomy	64
5.2.1	Visualizing taxonomy of <i>RNA</i> -family members	64
5.2.2	Comparing taxonomy of <i>RNA</i> -family members	65
6	Paper: RNALien - Unsupervised RNA family model construction	69
7	Paper: CMCompare webserver: comparing RNA families via covariance models	111
8	Discussion and outlook	117
	Bibliography	123

List of Figures

2.1	<i>RNA</i> nucleotides	6
2.2	Canonical base-pairs of <i>RNA</i>	8
2.3	<i>RNA</i> secondary structure motifs	10
2.4	Covariance of base-pairs	11
2.5	Types of gene homology	14
2.6	Phylogenetic tree and Cladogram	16
2.7	Taxonomic tree	20
2.8	Sequence alignment similarity matrix	24
2.9	Sequence alignment trace-back	25
2.10	Hidden Markov model	31
2.11	Protein family model	34
2.12	<i>RNA</i> -family model guide-tree with consensus sequence and structure of the XIST_A_REPEAT <i>Rfam</i> family	37
2.13	<i>Rfam</i> database family number development	40
3.1	Sequences per family in <i>Rfam</i> seed and full alignments	49
3.2	Structure conservation index (<i>SCI</i>) of <i>Rfam</i> family groups	50
3.3	Mean sequence identity (<i>MSI</i>) for <i>Rfam</i> family groups	53
3.4	Mean sequence identity (<i>MSI</i>) vs structure conservation index (<i>SCI</i>) of <i>Rfam</i> family subsets	53
5.1	Trackhub for predicted splicosomal <i>RNAs</i> in <i>Homo sapiens</i>	61
5.2	Assemblyhub for predicted <i>tRNAs</i> in <i>Escherichia coli</i>	63
5.3	Taxonomy of model organisms	66
5.4	Interactive taxonomic tree	67
5.5	Comparing taxonomy distribution of <i>CRISPR-Cas</i> systems in <i>Haloarchaea</i>	68
8.1	Workflow for model construction and evaluation	118
8.2	Specificity of <i>RNAlien</i> with BLAST query soft-masking	119
8.3	Recall of <i>RNAlien</i> with BLAST query soft-masking	120

List of Tables

2.1	Secondary structure representation of <i>RNA</i>	9
2.2	Taxonomic ranks	17
2.3	Overview of <i>RNA</i> groups	18
2.4	Algorithms for hidden Markov models	33
2.5	Algorithms for stochastic context free grammars	35
2.6	Covariance model guide tree nodes	36
2.7	Covariance model states	38
3.1	Structure conservation index for Rfam subsets	51
3.2	Mean sequence identity (<i>MSI</i>) for Rfam family groups	52
5.1	<code>track_hub_constructor</code> parameters	60
5.2	<code>assembly_hub_constructor</code> parameters	62

1 Foreword

I have always felt that the research of *RNA* holds amazing challenges, which has inspired me throughout my work in this field. In my diploma thesis I investigated *RNA-RNA* interactions (Tafer et al., 2011) in bacterial cells (Eggenhofer et al., 2011; Eggenhofer, 2011) and was confronted with several factors that impeded progress in unraveling these mechanisms. First, for a wide range of relevant — for example pathogenic — species, genomic annotation information was incomplete. In the case of pairwise *RNA* interactions, missing gene annotations mean that the corresponding interactions could not be predicted. Second, annotation of the genome has different degrees of sophistication. Poor annotation contains only the loci of a few protein-coding regions and assigns some hypothetical functions of these. Annotation can be extended with loci of non-coding *RNAs*, transcripts, repetitive regions and more. Such detailed genomic annotation is available for model organisms and closely-related species. Even if genomic features are present, however, annotations often comprise of just the protein-coding regions of genes without 3' and 5' untranslated regions. Such incomplete annotation means that the parts of the gene that could potentially interact, or inhibit interactions, are missing. In fact these deficiencies in accurate gene annotations not only affect interaction predictions but genome-scale analyses in general. A straightforward way to overcome this predicament is to improve annotation. Considering the large and ever-growing number of sequenced genomes, it is obvious that only automatic solutions would be able to have a significant impact.

At the time of my diploma thesis, two of my collaborators, namely Christian Höner zu Siederdisen and Ivo Hofacker, were working on the comparison of *RNA*-family models. These *RNA*-family models can be used to annotate additional instances of *RNA* genes from this family in other organisms. The application of *RNA*-family models is currently the most sophisticated method to improve the annotation for non-coding *RNAs*. While *RNA*-family models are available for established *RNA* families, it is not trivial to build new *RNA*-family models for novel *RNAs*. 2,473 of such *RNA*-family models are available via the **Rfam** (12.1) (Nawrocki et al., 2014a) database, which is the

most popular *RNA*-family database to date. However, there is a huge potential for additional *RNA* families that could be added to the pool and then used to improve annotation. There are 20,313 protein coding and in total 25,180 genes annotated for *Homo sapiens* in *Ensembl* (Yates et al., 2016) version 84. While some of the non-coding *RNAs* are unique to human, there will be many that have homologs in, for example other *mammalia*. Compared to the 2,473 families in the **Rfam** database, there is potentially an order of magnitude, or even more, additional families that could be entered only by taking the human genome into consideration.

Instead of starting with a project to automatically construct such families right away, I was given the opportunity to join the project for comparing *RNA*-family models. This was very useful to develop a understanding for problems in the use and construction of *RNA*-family models. Only afterwards I started to investigate *RNA*-family model construction itself. This actually is the reverse order of events as presented in my thesis, but I felt it is beneficial for the reader to first consider the construction of *RNA*-family models and then their evaluation.

2 Theoretical Background

RNA is at the center of some of the most essential biological processes. Foremost among these is gene expression, turning the genotype into a phenotype. The genotype, or heredity information, is encoded as Desoxyribonucleic acid or *DNA* and then transcribed into *RNA*. *RNA* viruses (Steinhauer and Holland, 1987) are an exception, which also use *RNA* as genome.

RNA genes that originate from a common ancestor gene, also called homolog *RNAs* can be found in different organisms. If a specific *RNA* has been investigated in one organism the gained insights are often also valid for these related *RNAs*.

This enables a knowledge transfer, which avoids redundancies and simplifies the planing of further experiments. A set of such homolog *RNAs*, that share the same biological function, is defined as *RNA-family*.

RNA-family models, describe the sequence and structure of a *RNA-family*. They enable the identification of additional members of a *RNA-family* via computational means.

This chapter presents the background and terminology for *RNA* biology and for *RNA-family* models.

The biological background describes *RNA* via its structure and function. Systematic approaches to define relationships between *RNA* genes, as well as their host organisms are introduced.

Construction of *RNA-family* models depends on sequence alignment. Moreover sequence alignment has inspired methods in homology search. Therefore the background includes a presentation of pairwise and multiple sequence alignment.

Dynamic programming, which is a method to solve complex problems by splitting them into sub-problems, is introduced alongside sequence alignment. However its use is ubiquitous in the field of bioinformatics.

RNA families are described via probabilistic models. First the simpler stochastic regular grammars and hidden Markov models that can be used to model primary sequence information are introduced.

However secondary structure information of homolog *RNAs* is often conserved

when no sequence similarity can be detected. Therefore the more complex stochastic context free grammars and *RNA* family models are presented, which are able to model secondary structure information.

The goal of providing automatic construction of these *RNA*-family models depends on the preexisting infrastructure. The **Infernal** (Nawrocki et al., 2009; Nawrocki and Eddy, 2013) toolkit includes programs to construct and process *RNA*-family models. *RNA*-family models are then archived in the curated **Rfam** (Nawrocki et al., 2014a) database.

The identification of homolog *RNA* genes, or homology search, via *RNA*-family models and alternative methods conclude the background chapter.

2.1 RNA biology

Ribonucleic acid or *RNA* is involved in many biological processes and potentially one of the most ancient components (Gilbert, 1986; Robertson and Joyce, 2012) of the cell.

RNA is a polymer formed of a sequence of different nucleotide monomers, which in different number and combination can serve different biological functions.

Despite their complexity, it is possible to describe *RNA* molecules via abstractions, also called primary, secondary, tertiary and quaternary structures (Alberts et al., 2002). These represent specific facets of *RNA* in different levels of detail.

RNA works as a means of information storage and transfer for cells and viruses. Moreover *RNA* can act as an enzyme and catalyze chemical reactions.

The diversity of *RNA* molecules we can observe nowadays arose from evolution. Understanding these evolutionary processes also allows us to find *RNAs* that originate from a common progenitor *RNA*, also known as homolog *RNAs*.

Homology can be considered in context with the host organism the *RNA* has been found in. Phylogeny and taxonomy are focusing on different features to put organisms in relation to each other.

Beyond sharing a common ancestor there are also other criteria to group *RNAs*, like *RNA* families, clans and classes. *RNA* families for example group *RNAs* via their function.

2.1.1 Sequence

Biological macro molecules like *DNA*, *Proteins* and *RNA* are polymer chains that consist of several specific types of monomers. *RNA* is composed of nucleotide monomers.

Each of these nucleotides consists of a nucleobase, a pentose sugar (ribose) and a phosphate. There are several canonical nucleobases that are generally part of *RNAs*, adenine, cytosine, guanine, uracil. They are abbreviated with the letters A,C,G,U (Comm, 1970). Guanine and adenine structures are based on a purine and therefore also known as purine bases, while cytosine and uracil are based on pyrimidine and known as pyrimidine bases.

There are other nucleobases (Helm, 2006) or modified versions of the four bases mentioned above. They are relevant in special cases, such as making *RNAs* resistant against degradation. In addition to naturally occurring nucleobases, artificial ones have been investigated (Leconte et al., 2008), as well.

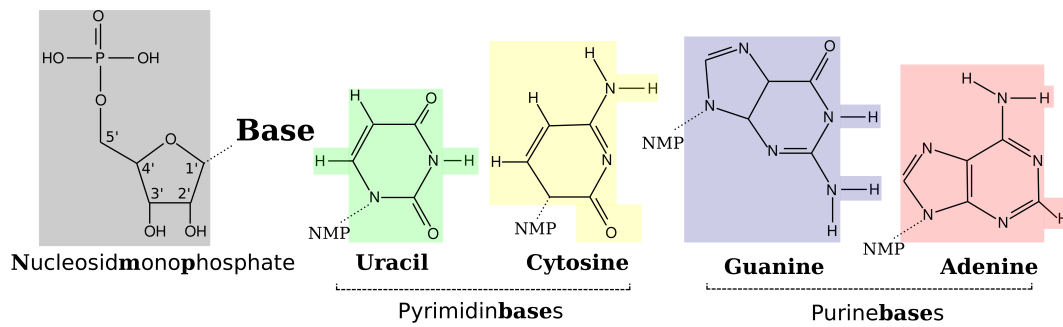


Figure 2.1: *RNA* nucleotides adopted from Eggenhofer (2011): The structural formula of a nucleosidmonophosphate is shown on the left hand side of the figure, which has a covalent bond to a nucleobase indicated by 'Base'. On the right hand side of the figure these nucleobases are shown, grouped into *Pyrimidinbases* (*Uracil*, *Cytosine*) and *Purinebases* (*Guanine*, *Adenine*)

The series of covalently linked nucleotides of the *RNA* molecule is commonly referred to as sequence, but also known as primary structure. The primary structure is usually written using the one-letter abbreviation, e.g. "AUGC" for a very short *RNA* consisting of four nucleotides.

The sequence is directed according to the phospho-diester bond connecting two nucleotides. Index numbers assigned to the five carbon atoms of ribose serve to give a name to the direction. The phospho-diester bond is formed between the oxygen bound to the 5 prime (abbreviated with the character ') carbon of one nucleotide and the oxygen bound to the 3' carbon of the next nucleotide (see Figure 2.2).

Sequences of naturally occurring *RNA* molecules can range from a few nucleotides long small *RNAs* (*sRNA*) (Storz et al., 2011) to thousands of nucleotide long ribosomal sub-units (Brimacombe and Stiege, 1985) or long non-coding *RNAs* (*lncRNA*) (Kung et al., 2013). The sequence of *U6 snRNA* (Brimacombe and Stiege, 1985), a non-coding *RNA* that is part of the spliceosome (Will and Lührmann, 2011) is shown in Table 2.1.

2.1.2 Secondary structure

Secondary structure refers to base pair interactions between nucleotides of the *RNA* molecule.

Base pair interactions are weak chemical bonds and based on non-covalent interactions (Lee and Gutell, 2004).

They came to prominence with the discovery of the *DNA* double-helix (Wat-

son et al., 1953). The two strands of the helix are connected via base pairs between cytosine and guanine, as well as thymine and adenine. The base pair interactions in the double helix are referred to as canonical base pairs.

While individual base pairs are weak compared to covalent bonds, the overall binding energy between the strands increases with their number. Base pairs, in combination with each other, can have a major influence on the structure of a molecule.

However, many other types of base pair interaction exist in three dimensional structures. The planar representation of nucleotides, as used in the structural formula, allows to define interaction surfaces, also known as edges. Leontis and Westhof (Leontis and Westhof, 2001) categorised base-pair interactions by the two edges that are interacting.

Purine bases each have a *Hoogsteen*, a *Watson-Crick* and a *Sugar* edge, while pyrimidine bases feature a *C-H*, a *Watson-Crick*, and a 'Sugar' edge for possible interactions. Canonical edges, as in the *DNA* double helix form between Watson-Crick edges.

Non-canonical base pairs refer to all the remaining interactions between edges of bases and the sugar component of the nucleotide. These interactions are even weaker, but there are many different combinations possible. Secondary structure considering non-canonical base pairs is also known as 2.5D, or extended secondary structure (Leontis and Westhof, 2001).

Both complementary canonical base pairs have similar distances between the outermost C1-atoms. This allows the formation of anti-parallel *RNA* helices. But also non-canonical base pairs share geometric properties, which can be used to partition base pairs into isosteric subsets (Leontis et al., 2002).

One *RNA* molecule can assume many different structures, which are formed by different combinations of canonical and non-canonical base pairs, between the nucleotides of this *RNA*. This is referred to as structure ensemble.

A specific structure is more likely to occur in a population of *RNA* molecules, the stronger the sum of all interactions between its base pairs are. The energy needed to break these interactions has been measured by melting experiments (Mathews et al., 1999b, 2004; Turner and Mathews, 2009).

The biological function of a *RNA* molecule is closely linked to corresponding structures. The *U6 snRNA* (Brow and Guthrie, 1988), for example is a part of the spliceosome (Wahl et al., 2009). It acts as a recycling factor for ribonucleic particles (Raghunathan and Guthrie, 1998). *U6* structure is adapted to provide binding sites for its interaction partners, like *U4 snRNA* (Bringmann et al.,

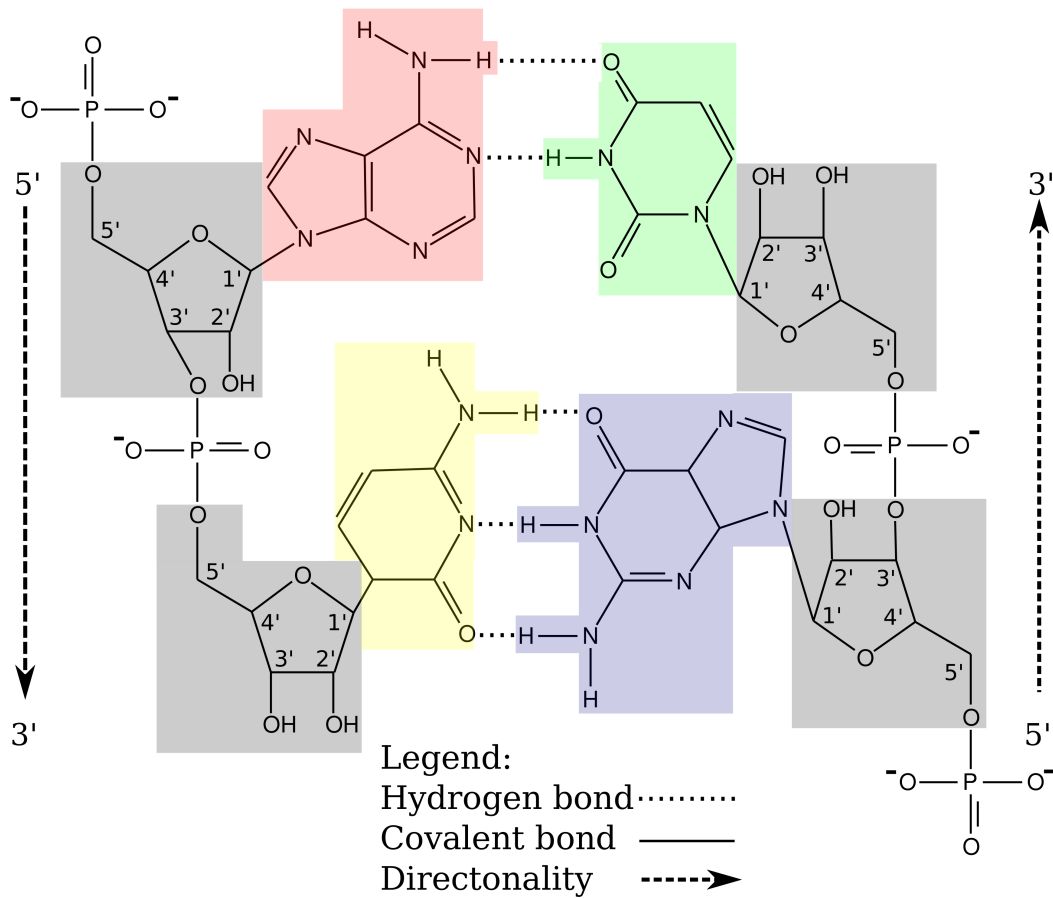


Figure 2.2: Canonical base-pairs of *RNA*, adopted from (Eggenhofer, 2011): Two anti-parallel strands of *RNA* are shown. The molecule on the left is oriented from 5' (top) to 3' (bottom) and consists of an adenine- (red) and a cytosine-nucleotide (yellow). The right hand molecule direction is reversed and starts with a guanine-nucleotide (blue) and ends with a uracil-nucleotide (green). The direction is defined by the labeling of the ribose (grey) carbon atoms. A canonical base-pair with two hydrogen bonds, between adenine and uracil is shown in the top-center of the figure, while a canonical base-pair between guanine and cytosine with three hydrogen bonds is depicted in the bottom-center.

1984) and *Sm* proteins (Hermann et al., 1995).

If the structure of the *U6 snRNA* changes, several other components of the spliceosome would need to adapt their structure to preserve the function (Cheng and Abelson, 1987). This constraint on structure can be observed for homolog *RNAs* that are far diverged, while the sequence of the *RNAs* might be only very weakly conserved.

Several basic reoccurring patterns can be observed in secondary structures formed by canonical base pairs. They can be combined and even nested

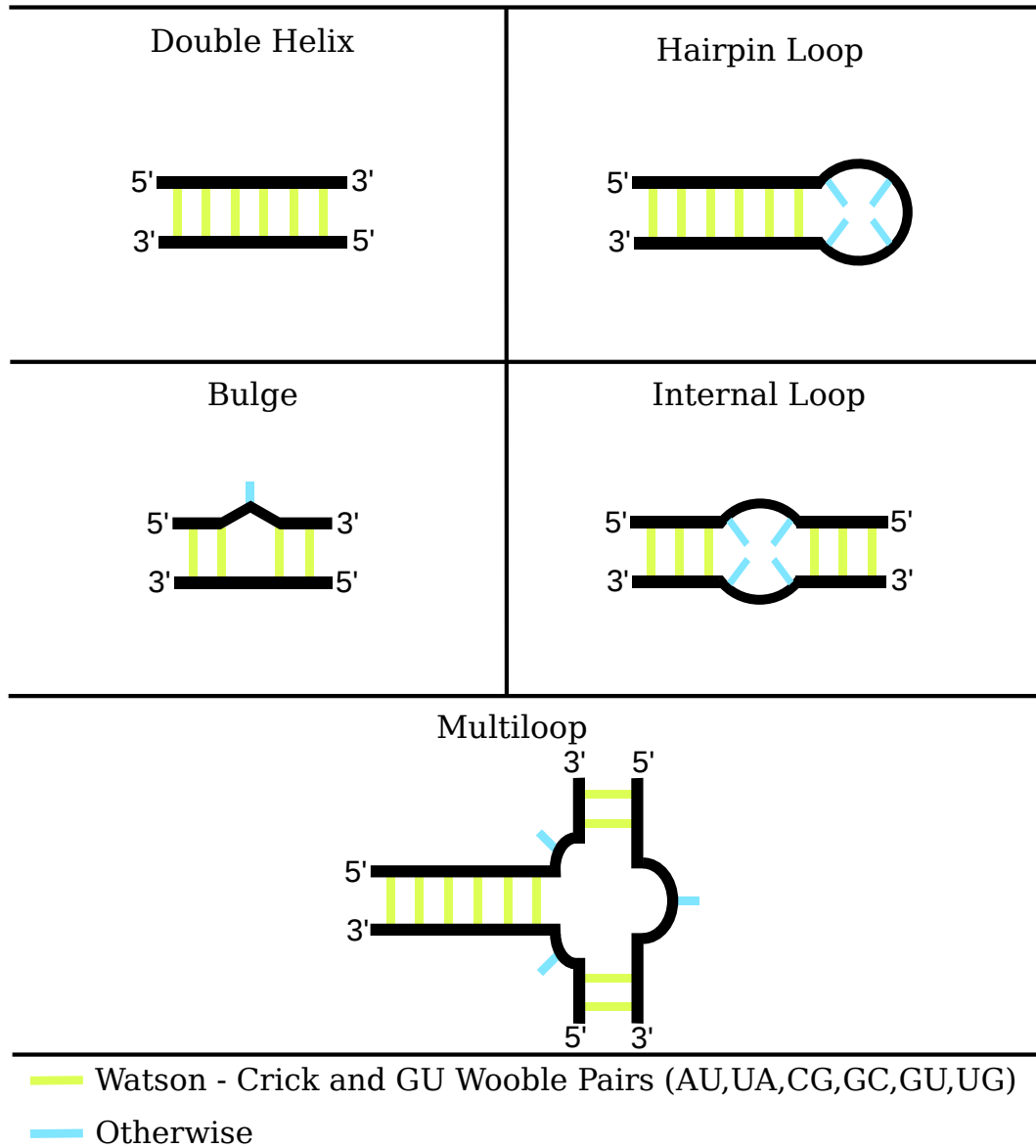


Figure 2.3: *RNA* secondary structure motifs, adopted from (Nowakowski and Tinoco, 1997) and (Gan et al., 2003): Black lines in the figure indicate the backbone, yellow lines show base pairs and blue lines unpaired nucleotides.

Atlas (Release 1.18). Secondary structure information can be augmented by annotating it with the location of known 3D-motifs (Petrov et al., 2013). Moreover tertiary structure themselves can consist of recurring tertiary structure building blocks, also called 3D modules (Hendrix et al., 2005).

A reason for this increased variability is that tertiary structure allows not only pairwise interactions between nucleotides. Nucleotide triplets occur in human telomerase where they stabilize a catalytically important pseudoknot (Kim

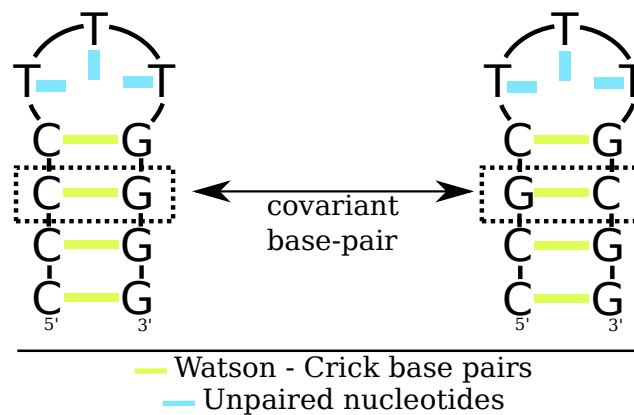


Figure 2.4: Covariance of base-pairs: The arrow indicates two covariant base-pairs. Mutations that disrupt biologically relevant structures can be compensated via a second mutation that restores the base-pair.

et al., 2008). Quadruplexes of nucleotides have been observed in ribosomal *RNA* Cheong and Moore (1992). A other major factor is coaxial stacking, that aligns helical regions in sequence to each other and is present in *tRNAs* (Quigley and Rich, 1976)

Tertiary structure motifs (Špačková and Šponer, 2006), as well as the modules they are composed of are frequently conserved between homolog *RNAs*. *U4 snRNA* features the kink-turn (Vidovic et al., 2000) motif. The kink turn is a common structural motif and can serve as a protein binding site (Schroeder et al., 2010).

A emerging approach is to consider secondary and tertiary structure in context with each other. 3D modules can guide prediction of secondary structure in a genomic scale and detection of genes (Theis et al., 2015).

2.1.4 Quaternary structure

Quaternary structure refers to interactions of the *RNA* molecule with other molecules. Binding sites on the *RNA* are specific for certain molecules or classes of molecules. Multiple binding sites can enable to formation of whole complexes of different molecules.

Therefor the knowledge about the binding partner for a *RNA* at all time points can be the key to understand the biological function of it. The interaction of a small bacterial *RNA* (Storz et al., 2011) can change the secondary structure or even the currently possible other binding partners of this *RNA*.

There are also much larger complexes, for example the spliceosome, in which *U6 snRNA*, introduced above, is participating. Depending on the function it

serves at this time point it can interact with *sm*-like Proteins (Vidovic et al., 2000), as well as *U4 snRNA* and others.

2.1.5 RNA function

Some *RNAs* are used as a template or message for translation into amino-acid polymers, also known as proteins. The two consecutive steps of translation and transcription have been defined as the central dogma of molecular biology (Crick, 1970). Besides these messenger *RNAs* (*mRNA*), there are *RNAs* that are not encoding proteins, called *non-coding RNAs* (*ncRNA*).

Transcription (Browning and Busby, 2004; Coulon et al., 2013) converts a gene from the genome into a *RNA* molecule, a transcript. This *RNA* has a sequence that is anti-parallel to the *DNA* it was transcribed from. Thymine, which does not occur in *RNA* is thereby replaced with Uracil.

There are several processes that are associated with transcription that can alter the *RNA* from being a reverse complement copy. Splicing (Matlin et al., 2005) removes certain parts of the transcript.

The parts that are cut out are referred to as introns, while the persistent parts are known as exons. Introns can, independently of their host transcript, fill biological functions (Westholm and Lai, 2011).

mRNA translation (Laursen et al., 2005; Dever and Green, 2012) converts nucleotide triplets of the transcript into a protein. The process of translation itself depends on several components that are composed of *RNA*.

The catalytic center of the ribosome (Palade, 1955) that is actually connecting amino-acid monomers is a highly conserved *RNA*, that is ubiquitous in the cellular organisms. This widespread presence in the tree of life combined with conservation (Ben-Shem et al., 2011) has been used as an argument that *RNA* is the most ancient component of life (Robertson and Joyce, 2012).

RNA is not restricted to catalyse the polymerization of amino-acids. *RNA* with catalytic properties is referred to as *ribozyme* (Kruger et al., 1982; Cech and Steitz, 2014).

RNA is involved in many other complex tasks, such as replication (Frouin et al., 2003), telomer extension (Bodnar et al., 1997), X-chromosome inactivation (Penny et al., 1996). They have conserved *RNAs* in common which will be introduced in the next sections.

2.1.6 Homology

Homology (Fitch, 2000a) of *RNA* and protein coding genes arises when they share a common ancestor gene. Homology can be divided into several sub types, which are described in the following text (see Figure 2.5).

Two types of events are relevant for homology. A speciation event is a process where one species develops into genetically different daughter species. The cause for this process can be mutations or recombination events. A duplication event is the duplication of a specific gene, which is afterwards present in two different loci in the same genome.

Ortholog genes result from speciation events. The gene is then present in two different species. In general they fulfill the same biological function.

Paralog genes are created through duplication events within one species. If the species is evolving into different daughter-species via a speciation event, two sub types of paralog genes can be defined. In paralog genes (Sonnhammer and Koonin, 2002) evolved by gene duplication after the speciation event. Out paralogs (Sonnhammer and Koonin, 2002) arise from duplications before the speciation event.

Paralog genes and gene duplications are highly relevant for evolution. One of the two genes is free to evolve into a new biological function, while the host organism retains the original function via the second gene. This process is known as functional divergence (Gu, 2003; Soria et al., 2014).

Xenolog genes originate from foreign (greek: ξένος, *xénos*) *DNA* that contains the gene and has been integrated in the cells genome. This process, also called horizontal gene transfer, can be mediated by viral/phagic vectors or by uptake from *DNA* from the extra cellular space. Horizontal gene transfer can be a driving force for rapid adaption to new environments, e.g. requirement of pathogenicity genes in bacteria (Gyles and Boerlin, 2013).

Genes with a similar phenotype, or function do not automatically result from the same genotype or even from homolog genes. The process which molds genes of different ancestry into the same phenotype is known as convergent evolution (Reece et al., 2011).

A big caveat for the annotation of homology is that genes that originated from convergent evolution are mistakenly annotated as homologs. This incorrect use of the term homology disregards the required evolutionary descent from a common ancestor gene (Marabotti and Facchiano, 2010).

Homology information makes it possible to share experimental data and annotations between organisms. There are numerous databases that provide

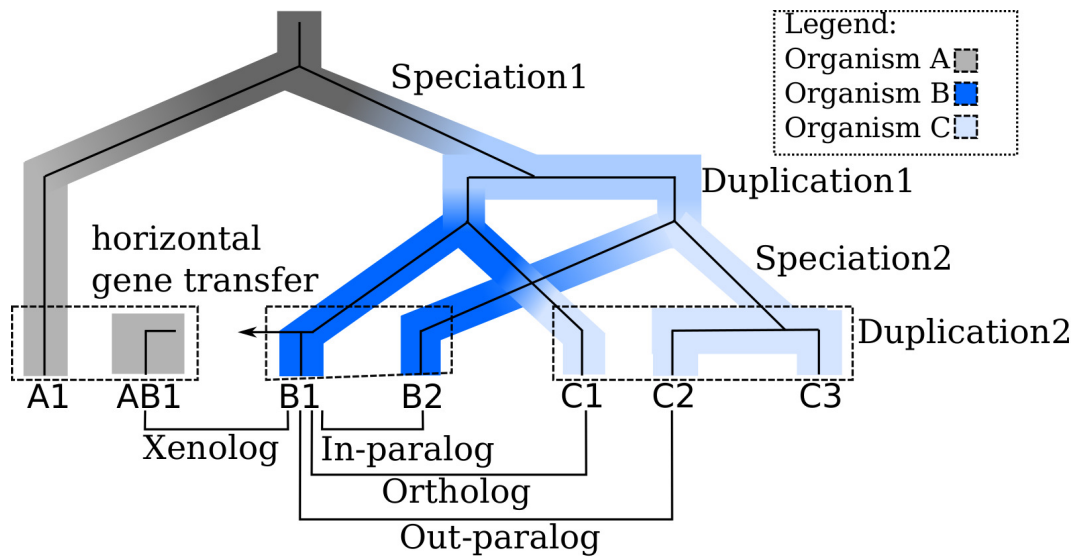


Figure 2.5: Types of gene homology: Homologs are genes that are derived from a common ancestor gene. There are 3 major types of homology. Orthologs arise from a speciation event and represent related genes in different species. Paralogs are created in gene duplication events, meaning that multiple copies of the gene are present in the organism. Xenologs originate from horizontal gene transfer, meaning the integration of DNA imported from outside of the cell in the genome. If the integrated DNA contains genes that share a common ancestor with one of the original genes, they are Xenologs. **A,B,C** denotes the daughter-species resulting from speciation events. Figure adopted from Fitch (2000b) and Richardson and Watson (2013)

this information (Huerta-Cepas et al., 2015; Wheeler et al., 2007) Homology of genes and their degree of conservation also influences how the ancestry of organisms is interpreted.

2.1.7 Phylogenetics

Phylogenetics describes the relationship between organisms and their common history in evolution. Heritable traits of the organisms like morphological properties or similarity of *DNA*, *RNA* and *protein* sequences is used.

Phylogenetics is based on the idea that all species share a common ancestor. A phylogeny of a certain group of organisms tries to trace the common paths of descent from their last shared ancestor.

The phylogeny is usually represented by a phylogenetic tree. This tree can be rooted, where the root represents the most recent common ancestor of all species in the tree. An unrooted tree just shows the relatedness of the included

species.

The leaves of the tree represent the actual species, while the branching pattern captures shared ancestors between them. The length of branches in the tree symbolizes the spent evolutionary time (see Figure 2.6 A).

Construction of phylogenetic trees by using traits is based on the phenotype, while using biological sequences is relying on the genotype.

The selection of traits that are evolutionary relevant is non-trivial due to several reasons. Some traits are very hard to measure, or even inaccessible to measurement, because only fossils remain of the species. The same phenotype can origin from entirely different genotypes by convergent evolution (Gaubert et al., 2005). Other phenotypes are so variable within one species that there is a strong overlap with other species (Swiderski et al., 1998).

Measuring the similarity between biological sequences is based on the distinct amino-acids or nucleotides of the bio-polymer. This is done via sequence alignment (see Section 2.2) and also due to the broad availability of sequence data currently the most-used method.

The mutations causing the difference between the sequences accumulated over a time. Therefore the amount of these mutations that makes two sequences different can be interpreted as a molecular clock. So the similarity of sequences can be interpreted as a evolutionary distances between species.

However the rate of mutations, or the speed of the clock, is depending on intrinsic factors like DNA repair systems and external factors, such as background radiation. These need to be considered to make evolutionary distances comparable.

Phylogenetic trees can be constructed via a variety of methods, e.g. distance based methods or maximum parsimony (Farris, 1970; Fitch, 1971). The former are using a matrix of pairwise similarity scores, which can be used by clustering algorithms. Subgroups of these sequence that are more similar to each other are clustered together.

The latter considers all possible trees that could be build with these sequences and selects the result tree by the minimal number of substitutions needed.

Cladistics is closely related to phylogenetics, but in contrast to it evolutionary time is not considered, but only the branching pattern of the lineages. It is represented by a cladogram (see Figure 2.6 B)

As mentioned above ribosomal *RNA* (*rRNA*) is one of the most ancient and widespread molecules in living organisms, with the exception of viruses. This makes it very useful the investigate relatedness between organisms.

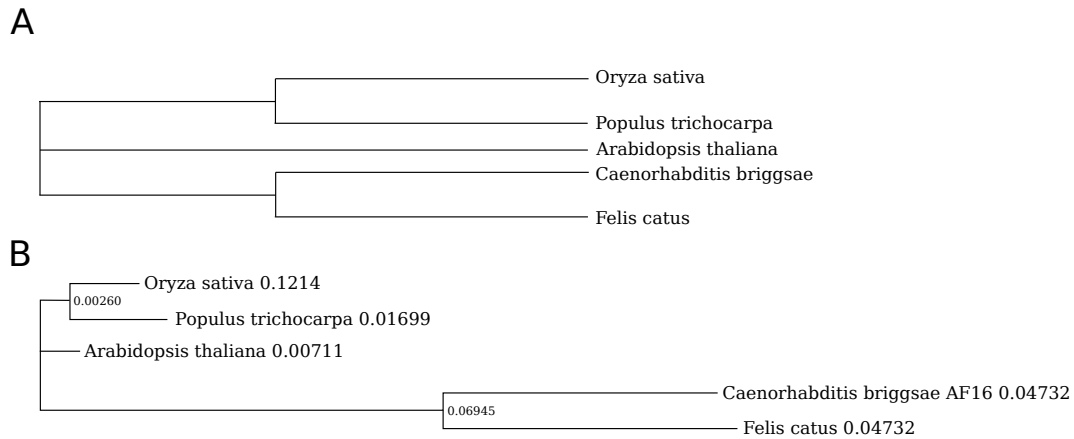


Figure 2.6: Phylogenetic tree and Cladogram: This figure shows a phylogenetic tree (A) and a cladogram (B) constructed for the *U6 snRNA Rfam* seed sequences from a *Clustal Omega* alignment. Each sequence is taken from a different species which is represented in the tree as leaves. The genetic distance of the sequences is expressed as horizontal edge-length in the phylogenetic tree. The distance to the next branching point, respectively the root is annotated for each leaf. The cladogram just shows the branching pattern and excluding the evolutionary distance. This figure is reproduced as a vector graphic from EBI webtools (McWilliam et al., 2009).

Components of the small ribosomal sub-unit are used, specifically 18 Svedberg *rRNA* for eukaryotic cells (Field et al., 1988) and 16 Svedberg *rRNA* for prokaryotic cells and mitochondria (Woese et al., 1990). Databases are available (Quast et al., 2013) that gather ribosomal *RNAs* and that allow to construct very comprehensive phylogenetic trees.

Such phylogenetic information can aid taxonomy, which is described in the next section.

2.1.8 Taxonomy

Taxonomy systematically puts organisms in relation to each other via shared traits. Modern taxonomy uses a classification scheme that was initiated by Linnean taxonomy (Linnaeus et al., 1758) along with binominal nomenclature. The binomial, or scientific, name consists of a first part, which denotes the genus and a second part that denotes the species. The house mouse for example belongs to the genus *Mus* and the species *Mus musculus*.

Taxonomy was since then influenced by phylogenetics but also by phenetics. Phenetics (taximetrics) (Sneath et al., 1973) constructs organism relationships from overall similarity and can be used to resolve species and subspecies clas-

sifications.

A rank-based system (see Figure 2.1.8) where the actual organisms are associated with the lowest ranks is used. Higher ranks are used to group organisms and the level of abstraction, meaning how general the used traits are, increases with the taxonomic level.

Table 2.2: Taxonomic ranks: Original ranks as introduced by Lineus with 2 examples. The rank phylum is used in a zoological context, while division is used in Botany.

Rank	Human	Maize
Kingdom	<i>Animalia</i>	<i>Viridiplantae</i>
Phylum/Division	<i>Chordata</i>	<i>Streptophyta</i>
Class	<i>Mammalia</i>	<i>Liliopsida</i>
Order	<i>Primates</i>	<i>Poales</i>
Family	<i>Hominidae</i>	<i>Poaceae</i>
Genus	<i>Homo</i>	<i>Zea</i>
Species	<i>Homo sapiens</i>	<i>Zea mays</i>

The traits used to divide organisms in different groups are very general features at the root of the tree and get more specific towards the species level, as opposed to the ranks.

Ranks, especially abstract ranks close to the root like Kingdoms, or Domains, are modified (Woese et al., 1990; Cavalier-Smith, 1998) or dropped due to ongoing discussions about the taxonomy system itself and related fields (Benton, 2000).

The rank system of the NCBI taxonomy database (Notabaart et al., 2005; Benson et al., 2008), is merged from many ranks used only in specific sub-taxonomies like plants or insects. This has the side effect, as clearly shown in Figure 2.7), that many ranks are not populated or taxonomic nodes are used that are not associated with a rank. The figure also shows that interior nodes are labeled, in contrast to phylogenetic trees or cladograms.

Despite huge efforts, taxonomy is still in flux and offers open challenges. There are constantly new organisms and traits that have to be considered and integrated in the existing system. Moreover making existing taxonomies compatible with each other results in artifacts, like different levels of coarse-graining as shown Figure 2.7).

However taxonomy is very useful due to the direct association of organisms in the taxonomy with genomic sequences and annotation. This information can also be used in the construction of RNA families. **RNAlien** uses taxonomy

information from the NCBI taxonomy database (Federhen, 2012), to search groups of closely related organisms for members of *RNA* families, see Chapter 6).

2.1.9 RNA groups

RNA families (Griffiths-Jones et al., 2003), *RNA*-family clans (Gardner et al., 2011) and *RNA* classes (Cech and Steitz, 2014) can be used to group *RNAs* that are shared between organisms. They are different in terms of biological function and homology, as shown in Table 2.1.9.

Table 2.3: Overview of *RNA* groups: *RNA* molecules can be grouped via different criteria like, sharing a common ancestor, sharing a biological function, the diversity of the family and if the members of the family are alignable by computational sequence alignment. Plus and minus symbols indicate the presence or absence of the property for the group in the same line or column. *RNA* families have common biological function and ancestry. *RNA*-family clans (Gardner et al., 2011) either share a common ancestor and function, but cannot be aligned due to the divergence of the family members (type 1, e.g. RNaseP (Ellis and Brown, 2009)) or they have clearly distinct functions, but are alignable and not divergent (type 2, e.g. *Glm* (Urban and Vogel, 2008)). *RNA* classes however have very generally the same function, indicated by the ~, but are not required to have a common origin.

<i>RNA</i> group	<i>RNA</i> -family	<i>RNA</i> clan type 1	<i>RNA</i> clan type 2	<i>RNA</i> class
Shared Biological function	+	+	-	~
Shared Ancestor	+	+	+	-
Diversity	-	+	-	+
Alignable	+	-	+	-

RNA families are sets of homolog *RNA* molecules, sharing a common ancestor and performing the same biological function in different organisms. Due to the close connection between biological function and structure, *RNA*-family members often share a common structure, while the sequence is much more variable.

RNA families are different in terms of in which organisms they are present and how divergent their members are. tRNAs for example are ubiquitous in non-viral genomes and can be very divergent, like missing one arm of the cloverleaf (Ohtsuki and Watanabe, 2007). Other families are restricted to only a very specific group of organisms and also highly conserved.

RNA-family clans Gardner et al. (2011) either share a common ancestor and function, but cannot be aligned due to the divergence of the family members (e.g. RNaseP Ellis and Brown (2009)) or they have clearly distinct functions, but are alignable and not divergent (e.g. *Glm* (Urban and Vogel, 2008)).

A *RNA* class is the group that is complementary to *RNA* clans. Its members have a common biological function in general terms, but are not necessarily derived from a common ancestor. The numerous known types of micro *RNAs* share their mechanism of action, but are not necessarily homolog.

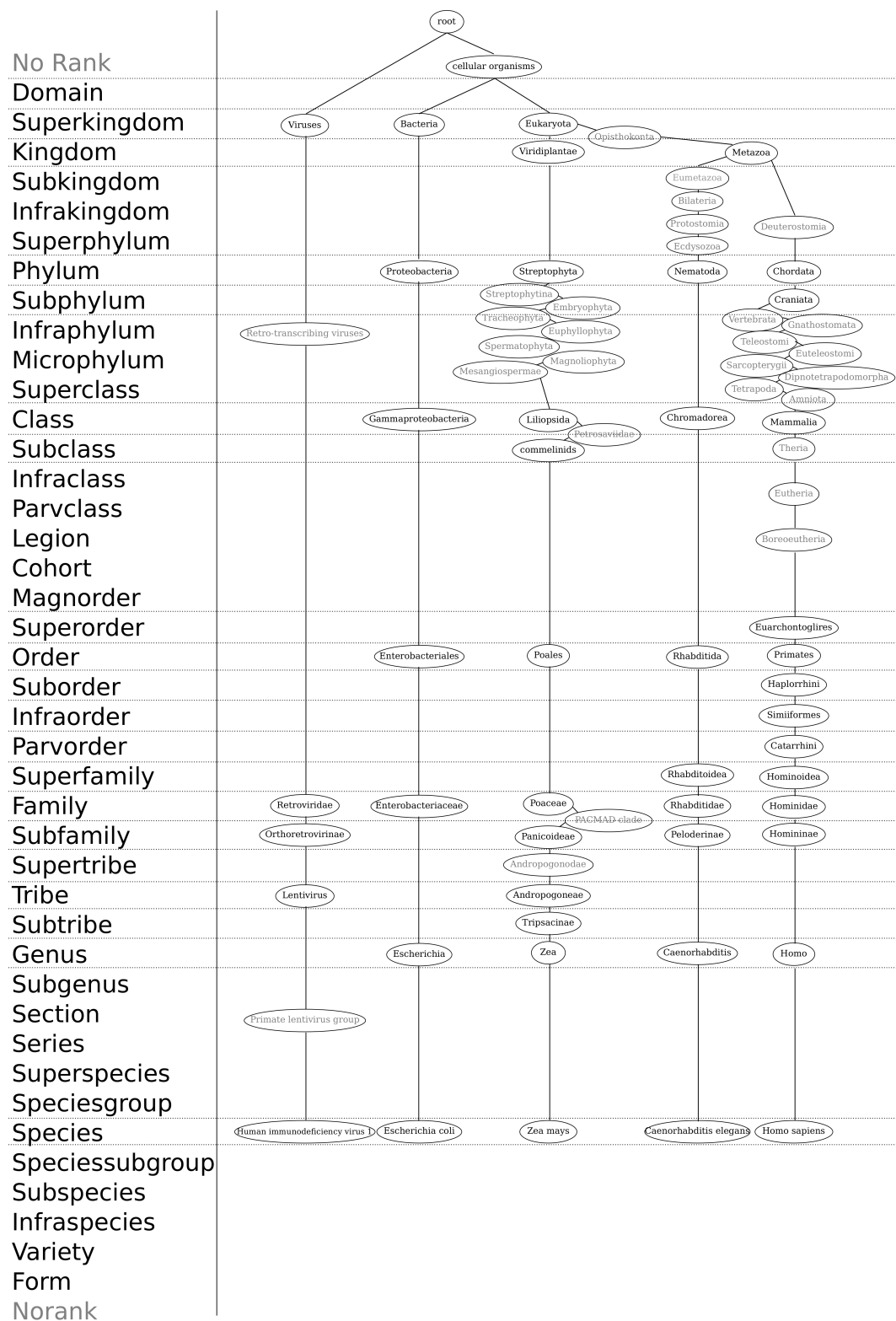


Figure 2.7: Taxonomic tree as used by NCBI (Notebaart et al., 2005; Benson et al., 2008), with example taxonomies for representative species from different kingdoms (*HIV1*, *Escherichia coli*, *Zea mays*, *Caenorhabditis elegans*, *Homo sapiens*). Taxonomic names are in the same line as their associated rank, those without rank are written in grey. The figure shows that many available ranks are not used, and on the other hand taxons are not associated with a rank.

2.2 Sequence alignment

Sequence alignment is a computational comparison of biological sequences. It matches regions of these sequences and reports their similarity. The similarity is useful for phylogenetic tree construction, as measure of evolutionary time that has passed. Moreover, high similarity can be an indicator for conservation and homology.

The computational representation of biological sequences is inspired by the one letter code of primary structure. The corresponding letters are encoded in a list-like data structure.

Sequences that share a common ancestor diverged due to mutation and recombination events. Alignment algorithms try to model these events based on the letter representation.

Point mutations can be interpreted as substitution of a letter, representing a monomer, with an other. Deletion events correspond to the removal of a character in one sequence, while insertions refer to the removal in the other sequence.

Sequence alignment methods are grouped by number of sequences they align with each other. The pairwise sequence alignment methods are discussed in special detail, with the intent to introduce the dynamic programming optimization strategy.

Dynamic programming is highly relevant for bioinformatics and has been used to make e.g. alignment, homology search and probabilistic model algorithms tractable.

2.2.1 Pairwise sequence alignment

The methods for pairwise sequence alignment either consider the alignment of the full length of both sequences (global), a part of one sequence against the full other sequence (semi-global) or only parts of both sequences(local).

Global alignment was first introduced by the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) and is used for evolutionary comparisons.

The similarity between two sequences can be determined by representing all possible pairwise-combinations of comparisons between the nucleotides as 2-dimensional array (Needleman and Wunsch, 1970).

A single comparison between the two sequences can now be treated as paths through this matrix. Some rules are needed for the step-wise moves to obtain meaningful results.

The first sequence x has n nucleotides and its i th position is represented by x_i , moreover the second sequence y has k nucleotides and its j th position is represented by y_j . Both indices are incrementing in 5' to 3' direction.

$A_{i,j}$ then represents the comparison between x_i and y_j in the comparison matrix. The alignment starts either from the first column or the first row. The path traverses the matrix by either incrementing the index of one or of both indices. Each pairwise comparison that we visit along the path is either a match, a mismatch, or a gap. The simplest form of scoring is to add 1 for each match along the path and 0 otherwise. The path ends when one or both indices are equal to the length of the sequence.

A naive approach for this problem would have exponential run-time, because we would recompute the values of all previous comparisons, every time we increment the indices.

Dynamic programming, or memoization makes this problem tractable by formulating the different possibilities for each step as a recursive function. A recursive function is calling itself until some termination condition is fulfilled, it then stops and returns an outcome. The outcome for each pairwise comparison is stored, or memoized in the matrix $A_{i,j}$.

With the simple scoring suggested above we would look for the path with the maximal score, indicating the highest number of matches. But this scoring does not consider similarity between nucleotides and does not punish deletions. This is a corresponding scoring function:

$$\sigma(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

A scoring function can be used to introduce such more complex scoring schemes. This function takes the characters that are compared as parameters and returns a score. A scoring function that treats substitutions between nucleotides of the same class (*Purine*, *Pyrimidine*) as neutral is shown in Equation 2.2

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a, b \in Y \\ 0 & \text{if } a, b \in P \\ -1 & \text{otherwise} \end{cases} \quad \text{where } Y = \{A, G\} \quad P = \{U, C\} \quad (2.2)$$

Alternatively the score can be interpreted as distance instead of similarity.

Matches do not influence the distance and are scored with 0, while substitutions, insertions and deletions are increasing the distance and contribute positive scores. In this case the alignment with the minimal score would indicate low distance and high similarity.

Gaps arise either from insertion events in one sequence or deletions in the other. The simplest model is to assume linear gap costs that corresponds directly to the number of gaps. If similarity is used in the scoring function and awarded with positive scores, the gap costs should decrease the score and set to e.g. $\gamma = -2$

Affine gap costs (Gotoh, 1982) are a more sophisticated model and assume that deletions and insertions often arise from recombination events, which cause multiple gaps at once. The model then punishes the initial gap severely, however the extension of the gap weakly.

Boundary conditions for the algorithm, also called initialization can be encoded in $A_{i,j}$. The neutral starting value for the global alignment $A_{0,0}$ is set to 0. The values stored in $A_{i,0}$ and $A_{0,j}$ represent the leading gaps in the alignment and increase directly with the length of the gap in case of linear gap costs setting $A_{i,0} = i * \gamma$ and $A_{0,j} = j * \gamma$.

A recursion for global sequences alignment with the Gotoh algorithm (Gotoh, 1982), adopted from Durbin et al. (1998), is shown in the following equation

$$A(i, j) = \max \begin{cases} A(i-1, j-1) + \sigma(x_i, y_j) \\ A(i-1, j) - \gamma \\ A(i, j-1) - \gamma \end{cases} \quad (2.3)$$

, where $\gamma = -1$ and σ as defined in Eq 2.1

The algorithm uses linear gap costs (γ) and has a asymptotic run-time (Knuth, 1976) and memory consumption of $\mathcal{O}(n * m)$, where n is the length of the first and m of the second sequence.

The resulting similarity matrix, for this recursion applied to two short sequences, is shown in the following Figure 2.8. The final value representing the best global alignment can be found in the bottom right cell of the matrix.

To obtain not only the score of the best alignment but the alignment itself a method called backtracking has to be applied. This reverses the computing of the scores and starts in the cell with the best score. Then for each step the path to the cell the current value was derived from is selected. If there is more than one equal way to arrive at the value, then there are that many alternative choices. In the standard variant of backtracking only one of these is selected by

A_{ij}		x_0	x_1	x_2	x_3	x_4	x_5	x_6
		-	A	G	C	U	C	C
y_0	-	0	-1	-2	-3	-4	-5	-6
y_1	A	-1	1	0	-1	-2	-3	-4
y_2	G	-2	0	2	1	0	-1	-2
y_3	C	-3	-1	1	2	1	1	0
y_4	C	-4	-2	0	1	2	2	2
y_5	C	-5	-3	-1	0	2	2	2

Legend:

Final score (best alignment) ■

Initialization values ■

Sequence indices ■

Figure 2.8: Sequence alignment similarity matrix: The matrix contains the tabulated intermediate results of the global alignment recursion. The indices of both sequences are indicated by the subscript of the text in grey. The score of the best comparison result for the two subsequences up to this point is shown in the corresponding cells. The initial values in blue indicate leading alignment gaps in one sequence. The final result in red can be found in the bottom right cell and is highlighted in red.

arbitrary choice. Therefore only one optimal alignment will be produced, but the algorithm can be modified to recover more than one alignment (Altschul and Erickson, 1986).

The selected paths determine the two characters that are added to the alignment. The diagonal move that gave rise the match or mismatch scores add the two corresponding nucleotides x_i and y_j to the alignment. Alternatively horizontal moves add the y_j nucleotide and a gap, while vertical moves add the x_i nucleotide and a gap. An example for the trace-back and the optimal alignment resulting from the previous example is shown in Figure 2.9.

Semi-global, glocal, free-end gap, or free shift alignment methods are used to align a shorter sequence to a longer one, for example aligning a sequence motif to a transcript (Brudno et al., 2003). This problem can be solved with a variant of the Needleman-Wunsch algorithm, where flanking gap-costs are not considered. The first column and row are therefore all initialized with zero.

Local alignment methods are useful if sequence share a similar domain, but are otherwise different. The Smith-Waterman algorithm (Smith and Waterman, 1981), is modified a version of the Needleman-Wunsch algorithm and was the first algorithm for this task.

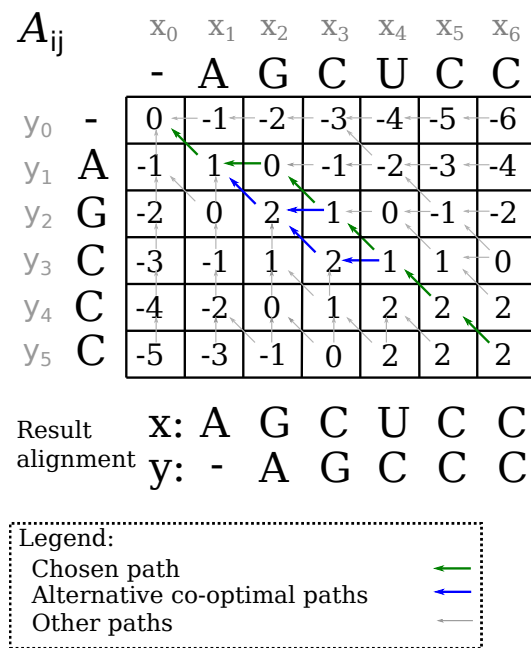


Figure 2.9: Sequence alignment trace-back: The alignment for the scores computed by the global alignment recursion can be obtained by trace-back. Starting from the result value in the right bottom cell, the alignment is constructed depending on which cases of the recursion yielded the score. Diagonal moves originating from match or mismatch cases, vertical or horizontal moves from indels (insertions or deletions). The match and mismatch case mean that the two nucleotides with indices corresponding to this field are added to the alignment. In the indel case, a nucleotide of one sequence and a gap character in the other are added to the alignment. It is possible that two alternative cases yield the same score, this means that there are two equally good alignments. The result alignment for the selected path is shown below the backtracking matrix.

Additionally to similar initialization as in semi-global alignment, the recursion features a fourth alternative to score a cell with 0. This indicates the start of a new local alignment.

Structural alignment

Secondary structure is often better conserved than the sequence information. This conservation allows to detect homology even when there is no sequence similarity left. Especially for *RNA* alignments, which do not have a reading frame available, like proteins, it is essential to incorporate structure information to improve alignment quality.

However for most *RNAs* a experimentally determined structure is not available.

Structural alignment algorithms therefore also have to solve the problem of missing structure information.

The Sankoff algorithm simultaneously computes the alignment and the consensus secondary structure of a sequence pair in $\mathcal{O}(n^6)$ and $\mathcal{O}(n^4)$, where n is the length of the sequences (Sankoff, 1985).

These substantial memory and runtime requirements have been addressed via two different variants of the Sankoff algorithm. The first one places restrictions on the aligned sequences by e.g. by defining maximal motif lengths, or maximal subsequence length differences. Examples for this approach are **Foldalign** (Gorodkin et al., 1997; Sundfeld et al., 2015), which was the first implementation that supported local alignment and **dynalign** (Mathews and Turner, 2002), which implements a full loop based thermodynamic energy-model (Mathews et al., 1999a).

The second approach restricts the base-pairs that are considered during the sequence-structure alignment. **pmcomp** (Hofacker et al., 2004) first computes the base pair probability matrices for the individual sequences via the McCaskill algorithm (McCaskill, 1990; Hofacker et al., 1994; Lorenz et al., 2011). The pairing probabilities are used to score the pairing between nucleotides during alignment. By restricting the span between matching base pairs for all partial alignments, the runtime complexity is reduced to $\mathcal{O}(n^4)$ and the memory consumption to $\mathcal{O}(n^3)$.

LocARNA (Will et al., 2007, 2012) improves this approach and also applies it to further lower the memory consumption. By assuming a minimal probability threshold for base-pairs, the necessary matrix becomes sparse and only requires $\mathcal{O}(m^2 + n^2)$ memory and $\mathcal{O}(n^2(m^2 + n^2))$ time, where n is the sequence length and m the number of significant base pairs. **RNAlien** (Eggenhofer et al., 2016) (see Chapter 6), uses **LocARNA** for the semi-global alignment of *RNA* fragments, found in homology search, to a full length homolog *RNA*.

2.2.2 Multiple sequence alignment

Multiple sequence alignment (*MSA*) algorithms perform comparisons between three or more sequences. There is a wide range of applications for *MSA*. Evolutionary distances between a set of sequences as required for phylogenetic trees can be computed. Regions that can be aligned over multiple sequences can indicate conserved domains that are relevant for the biological function.

Most relevant, in the context of this thesis, is that multiple sequence alignments are the prerequisites for *RNA*-family model construction. The quality of the

alignment thereby directly affects the quality of the resulting model.

Multiple sequence alignment is a difficult problem, which has lead to several approaches. The field has been evolving for some time and some of these methods build on others, like iterative alignments on progressive ones.

Exact methods

Like pairwise sequence alignment, also multiple sequence alignment could be solved with dynamic programming. The number of dimensions of the dynamic programming matrix would then correspond with the number of sequences. A exact algorithm for this problem (Sankoff, 1975) exists. The run-time and memory requirements for sequences of same length (\bar{L}) are $\mathcal{O}(\bar{L}^N)$ and $\mathcal{O}(2^N \bar{L}^N)$ (Durbin et al., 1998). This is not practical for sets of longer sequences beyond three.

However there is a range of heuristic methods that not necessarily give the best solution for the problem, but are much faster.

Progressive alignments

Progressive alignment methods utilize pairwise alignment of the input sequences. Starting from a first pair the other sequences are step-wise added to the growing alignment, which is fixed after each step.

The sequences are generally added in order of their similarity. The progressive alignment approach introduced by Feng and Doolittle (Feng and Doolittle, 1987) first computes all pairwise similarities. Then a guide tree is constructed by clustering (Fitch et al., 1967). The guide tree represents the ordering in which the sequences are going to be aligned. It is a binary tree where the leaves represent the input sequences and the interior nodes represent alignment steps. The outermost leaf nodes that represent the most similar sequences are aligned first. Then the other sequences are aligned to the fixed alignment until the leafs are exhausted.

CLUSTALW (Thompson et al., 1994) takes the position-specific sequence conservation (Gribskov et al., 1987) of the already partially aligned sequences into consideration. Sequences are then progressively aligned to this profile and the nucleotide frequencies modified accordingly.

T-Coffee (Notredame et al., 2000), tries to avoid the uses a library of sub-alignment information. In the first step pairwise alignments between the input sequences are computed, also known as primary library. T-Coffee has the

advantage that it can in principle use any pairwise sequence alignment tool to build the primary library.

The contribution of different sub-alignments used (e.g. global or local) can be controlled by assigning weights to them. An extended library is constructed from the edges of the primary library. For all pairwise alignments in the primary library the alignments with a third sequence are used to determine the consistency of edges. The result is an extended library that contains modified edge weights corresponding to this consistency information. The extended library is then used for scoring in the progressive alignment.

Due to their heuristic nature and the propagation of errors from previous alignment steps to the final result the quality of progressive alignments is often not optimal. Other methods have been conceived to improve upon progressive alignments.

Iterative alignments

Iterative alignments methods are refining existing alignments by realigning. This allows to improve the alignment of first aligned sequences that were fixed from the beginning of the alignment process. A strategy to achieve realignment is to remove aligned sequences from the alignment and realign them to the profile of the other sequences. This process is iteratively repeated until the alignment scores converge. Examples for iterative alignment methods are MUSCLE (Edgar, 2004) and DIALIGN (Morgenstern, 2004).

Multiple structural alignments

The fact that secondary structure information is often better conserved than sequence information is also essential for the quality of multiple *RNA* alignments.

The Sankoff algorithm could in principle also be used for multiple sequence alignment. The runtime complexity is $\mathcal{O}(n^3 K^N)$, where n is the length of the longest sequence, K is a small integer constant proportional to n and N the number of sequences.

PMcompMulti (Hofacker et al., 2004) was the first tool using the progressive alignment strategy in multiple structural alignment.

mLocARNA (Will et al., 2007, 2012) was initially also a progressive multiple alignment method. Since then it has been extended to provide also iterative and consistency based multiple alignments.

It relies on the pairwise alignment algorithm **LocARNA** (Will et al., 2007, 2012) to perform both local and global structural alignments.

RNAlien (Eggenhofer et al., 2016) uses **mLocARNA** for computing the multiple sequence alignment from which the initial *RNA*-family model is constructed (see Chapter 6).

A alternative method is available via the multiple alignment tool **FoldalignM** (Torarinsson et al., 2007; Havgaard et al., 2012).

2.3 Probabilistic models

Hidden Markov models (*HMM*) and stochastic context free grammars (*SCFG*) are the probabilistic models that have been most commonly applied to represent sets of *RNA* sequences. *HMMs* will be presented first and *SCFGs* explained as an extension of them.

2.3.1 Hidden Markov models

Different concepts and terminologies have been associated with *HMMs* due to their versatility and use in a different fields. They can for example also be understood as stochastic regular grammars.

Originally Hidden-Markov models were applied in speech recognition (Rabiner, 1989), but they were also used in electrical engineering (Satish and Gururaj, 1993) to classify discharge patterns.

HMM are based on the Markov process which describes transitions between states of a finite state space. In a stochastic Markov process each of these transitions is associated with a probability.

A process that is a Markov process must fulfill the Markov property of being memory-less. For a first order *HMM* the conditional probability of the current state only depends on the previous state. Higher order hidden Markov models of rank n depend on the n previous states.

The 'hidden' property of *HMM* means that only specific states, produce visible output with a specific emission probability.

The model consists of states, emissions and probabilities. A *HMM* can be defined as a tuple $(S, \Sigma, (\pi_i), (a_{i,j}), (b_{i,k}))$ where S is a set of states $1, \dots, n$, Σ is a set of emittable symbols (alphabet) $1, \dots, m$, π are the starting probabilities, $a_{i,j}$ are transition probabilities and $b_{i,k}$ are emission probabilities.

An example model for the feeding call behavior (Evans and Marler, 1994) of male *Gallus gallus* specimen, using the introduced notation is shown in Figure 2.10.

A traversal of the model following always one edge from a state into the direction indicated by the arrow would yield emissions which corresponds to the calls emitted by the bird.

Design of a new model first considers possible states, emissions and allowed moves between states. In the second step probabilities are assigned, which can be subject to further modification.

The model has four states, that represent the start of the observation (S),

if food is present for the observed specimen, or not (N) and the end of the observation (E). While (S), (F) and (N) are known as non-terminal states, (E) is a terminal state and terminates the traversal.

Beginning with the starting state the model proceeds with a starting probability to either an immediate end of the observation, for example if there is no specimen to observe, or to the (S) or (N) state.

These two hidden states can produce emission with certain probabilities. The food present state has a higher probability to yield a food call (C) instead of other sounds (O) compared to the no food present state. After the emission the model can proceed to either again to or (N) or (F) or to (E).

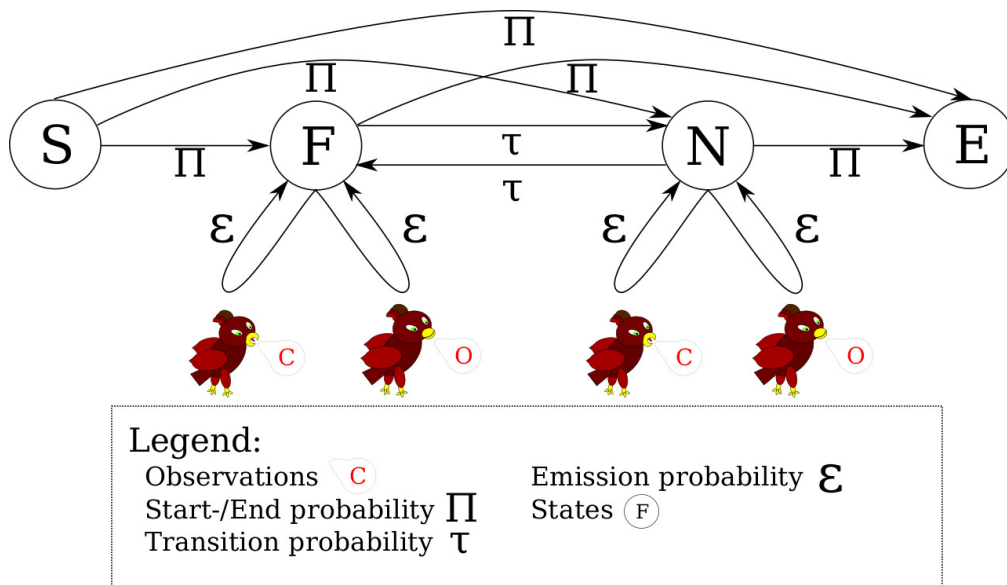


Figure 2.10: Hidden Markov model example for modeling the feeding call. The model consists of states S (Start), F (Food), N (No food), E (End); a output alphabet C (food call) and O (other sounds or silence) in red; starting and ending probabilities Π ; transition probabilities τ and emission probabilities ϵ .

Once the model has been set up, different algorithms are available to answer questions pertaining to the model and what it represents. Three of these algorithms and their goals are presented below.

Given a series of observations there is the question which is the best path of states in model. Applied to the bird call model this would mean which sequence of food present and no food present most probably produced the bird sounds.

The most likely path through the states and therefore the most likely sequence of emission can be computed by the Viterbi algorithm (Viterbi, 1967). This

sequence of observations can be produced from multiple state sequences. The algorithm selects the path of maximal probability for each time-step.

The second question is determining the probability of a certain observation given the model. Translated to the feeding call model this would be series of food calls and other sounds. An actual series of bird calls should therefore achieve a high probability, while a bad imitation should have the opposite result.

The forward algorithm computes the probability of a series of observations given the model and its parameters. Instead of selecting the path of maximal probability, the forward algorithm sums the probability for all paths that can produce the observation from the start up to a certain time point. Formulated differently that gives the probability of a certain prefix of observations given that the *HMM* was in a specific state at that time-point.

The backward algorithm computes the probability of these sequence given the model and its parameters. In contrast to the forward algorithm, the backward algorithm starts at the last time-step and sums the probability of all paths that can produce a certain observation up a certain time-point, backward in time. Formulated differently that gives the probability of a certain suffix of observations given that the *HMM* was in a specific state at that time-point.

In most cases the parameters, or probabilities in the model, especially the movements between hidden state can not directly be observed as in the feeding call model. Instead expectation maximization (Dempster et al., 1977) is used to iteratively improve parameters with the forward-backward (Baum, 1972) algorithm.

A naive approach in computing answers for these three problems would take exponential run-time and memory consumption, because considering all possible traversals of the state-space the same results would be computed over and over again.

The Markov property allows us to use dynamic programming to achieve polynomial run-times. Intermediary results are memoized and do not have to be computed multiple times (see Table 2.4).

As suggested above hidden Markov models can also be perceived as stochastic regular grammars. The concept of grammars is used in linguistics and theoretical computer science (Chomsky, 1959).

A grammar is defined by rewrite, or production rules and symbols. Symbols can be split into non-terminal and terminal symbols. In context of hidden Markov models, terminal symbols could be interpreted as states and terminal

symbols as the emitted observations.

The production rules have a left-hand, containing at least one non-terminal symbol and a right hand side, containing terminal and non-terminal symbols. By applying the production rules a string of non-terminals is generated.

A grammar is stochastic, when the emission of non-terminal symbols and the selection of production rules are only happening with a certain probability.

Regular grammars allow productions in the form of $W \rightarrow aW$ or $W \rightarrow a$, where W can be any non-terminal symbol and a any terminal symbol. This is sufficient for the primary structure of bio-polymers.

Table 2.4: Algorithms for hidden Markov models (*HMM*), adopted from (Durbin et al., 1998). The runtime complexity of these algorithm is $\mathcal{O}(LM^2)$ and the memory consumption $\mathcal{O}(LM)$, where L is the length of the observation and M is the number of states

Goal	HMM algorithm
Optimal alignment	Viterbi
Probability($s \Theta$)	Forward
Expectation maximization parameter estimation	Forward-Backward

Hidden Markov models were very successfully applied to describe Protein families. These Protein family models are constructed from multiple sequence alignments and capture which regions of the protein are conserved and where gaps or insertions are occurring.

The models use begin, match, insertion-deletion (indel) and end states. The occurrence of a specific state is modeled via transition probabilities. The frequency of a specific amino acid at a certain position is encoded in emission probabilities. An example for a toy protein *HMM* is shown in the following figure 2.11.

Tools and databases for protein family models are briefly mentioned, because they inspired a similar infrastructure for *RNA* families.

The *HMMER* (Eddy, 1998, 2011; Mistry et al., 2013) toolkit contains programs to construct hidden Markov models for proteins and to use them for homology search.

The curated **Pfam** (Bateman et al., 2004; Finn et al., 2013) database exists that collects these models and their corresponding multiple sequence alignments.

The independent columns of a *HMM* are well suited to model the primary structure of proteins or *RNA*. However the importance of using secondary structure information for detecting homology between *RNAs*, makes it necessary to include base-pairing information. This introduces dependencies be-

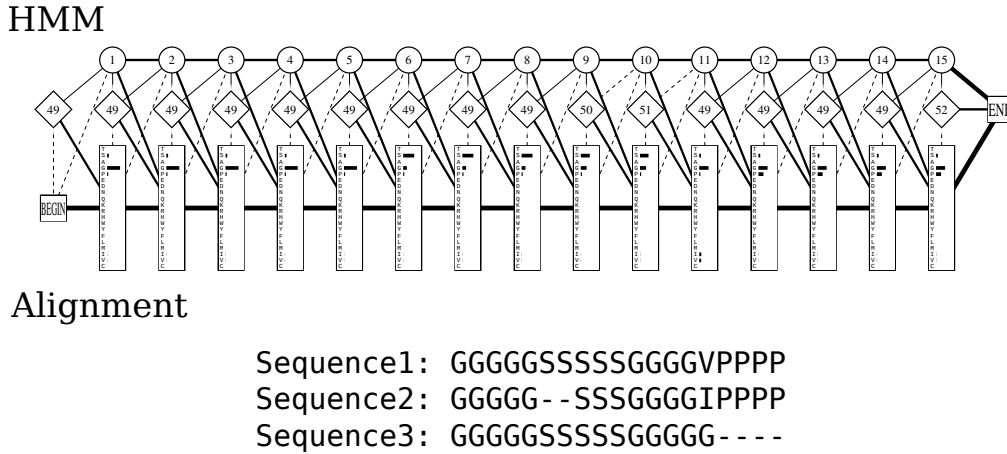


Figure 2.11: Example for a protein family model built and visualised with **SAM** (Krogh et al., 1994; Hughey and Krogh, 1996) from the toy alignment that is included in the figure. The figure show begin and end states on the left, and right end of the model. Indel states are round and shown at the top of the visualisation, match states are shown in the center. Transition probabilities connect the state as edges. The higher the probabilities of these edges, the thicker the plotted line. Emission probabilities for all amino-acids are shown via the length of the bar next to the one-letter abbreviation of the amino acid in the box at the bottom of each position.

tween columns which cannot be modeled by regular grammars and has led to the use of context free grammars.

2.3.2 Stochastic Context Free Grammar

Stochastic context free grammars, or *SCFGs*, expand on the concept of stochastic regular grammars. They allow to model dependencies between remote states, as required for secondary structure in *RNA*-family models.

On the right hand side of the grammar, any combination of terminal and non terminal symbols is allowed. The corresponding production rule is $W \rightarrow \beta$, where W can be any non-terminal symbol and β any combination of terminal symbol and non-terminal symbols.

The less restrictive production rules allow more complex models, which for example can also model the secondary structure of *RNA*. While the goals remain the same, different algorithms are required for *SCFGs* (see Table 2.5). The problem of finding the best path for the sequence of observations is solved with the CYK algorithm (Sakai, 1962; Kasami, 1965; Younger, 1967; Cocke, 1970), which corresponds to the Viterbi algorithm for *HMMs*.

The probability of the observation given the model can be computed with the Inside algorithm (Lari and Young, 1990). Like the Forward algorithm it sums the probabilities of all possible paths for the sequence of observations.

The Outside algorithm (Lari and Young, 1990) is the *SCFG* equivalent to the Backward algorithm for *HMMs*.

EM parameter training for *SCFGs* can be solved with the Inside-Outside algorithm (Lari and Young, 1990, 1991), which corresponds to the forward-backward algorithm for *HMMs*

While *SCFGs* allow to build more complex models, they also require substantially higher computational cost in terms of run-time and memory. Depending on the sets of observation the model represents, it might be necessary to use cheaper approaches like *HMMs* or heuristics instead. This gain in performance induces a loss in descriptive power.

Table 2.5: Algorithms for stochastic context free grammars (*SCFG*), adopted from (Durbin et al., 1998). The runtime complexity of these algorithm is $\mathcal{O}(L^3 M^3)$ and the memory consumption $\mathcal{O}(L^2 M)$, where L is the length of the observation and M the number of states

Goal	SCFG algorithm
Optimal alignment	CYK
Probability($s \Theta$)	Inside
Expectation maximization parameter estimation	Inside-Outside

2.4 RNA-family models

Covariance models, commonly abbreviated as *CMs*, are profile stochastic context free grammars (Eddy and Durbin, 1994). These models feature the same properties as hidden Markov models, but the context free grammar also allows to model relationships between states that have longer distances. Homology search for *proteins* utilizes the sequence conservation that arises from the amino acid encoding codon triples. This is not possible for *RNA* homology search, however the secondary structure of *RNA* is often conserved and can be used instead.

Covariance models allow to include base pair interactions, but require an additional set of node types, compared to hidden Markov models that just model the sequence. The reference implementation of *RNA*-family models are used by the **Rfam** (Griffiths-Jones et al., 2003; Gardner et al., 2011; Nawrocki et al., 2014b) database and the **Infernal** (Nawrocki et al., 2009; Nawrocki and Eddy, 2013) tool package.

2.4.1 Infernal

RNA-family models are complex data-structures and the infrastructure for building and using them has grown over time. The **Infernal** package (Nawrocki et al., 2009; Nawrocki and Eddy, 2013), short for *INFERENCE of RNA ALIGNment*, is a central part of this infrastructure and contains ten major tools and and multiple scripts.

The consensus secondary structure is encoded in the guide tree of the model, which consists of nodes. There are eight different node types as shown in Table 2.6.

Table 2.6: Covariance model guide tree nodes: adopted from Infernal user guide (Nawrocki and Eddy, 2013)

Node type	Description	States
MATP	pair	MP, ML, MR, D, IL, IR
MATL	single strand, left	ML, D, IL
MATR	single strand, right	MR, D, IR
BIF	bifurcation	B
ROOT	root	S, IL, IR
BEGL	begin left	S
BEGR	begin right	S, IL
ROOT	end	E

An example guide tree for the XIST_A_REPEAT family is shown in Figure 2.12. The figure shows, how the ability of context free grammars to produce sequences from inside out is used to capture the consensus secondary structure of the family. The lines connecting the guide tree nodes to the consensus structure and sequence also visualize the meaning of left and right in the nodes names.

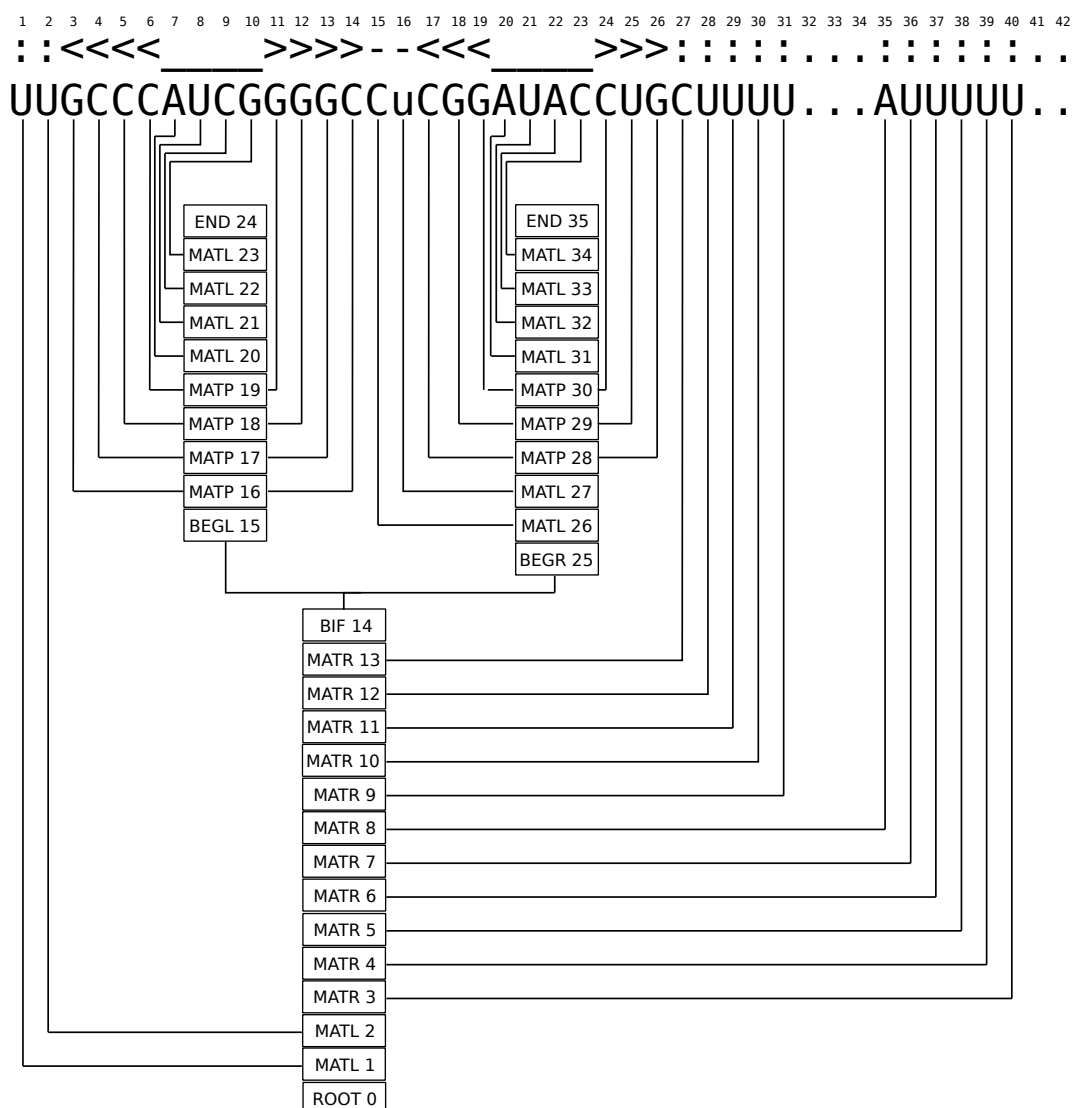


Figure 2.12: RNA-family model guide-tree with consensus sequence and structure of the XIST_A_REPEAT Rfam family.

The nodes can contain several states, which encode the sequence. Seven different states, see Table 2.7 are defined. Pair-emitting states (P) are used to model base pairs, left (L) and right (R) emitting states for nucleotides in matches and insertions. Depending on this context the corresponding variants are prefixed

with M, respectively I (see Table 2.6). Deletions are represented by the D state, end by E and start by S states. The bifurcation state B can be used to model nested structures.

Each of the nodes is associated with grammatical production rules. Which symbols are emitted by these productions and which state, if any, the grammar will transit to is controlled by emission and transition probabilities for each state.

Table 2.7: Covariance model states, adopted from Infernal user guide (Nawrocki and Eddy, 2013). A description for each state type and the grammatical production rules are tabulated. N refers to unspecified non-terminal symbols (the states P,L,R,B,D,S,E), the small letters x and y to unspecified terminal symbols (the nucleotides in letter code a,g,u,c). Small epsilon represents the empty string. The P state emits 2 nucleotides (x,y) , which delimit a new state N . As indicated in Table 2.6 some of the states can be used either in a match case or in insertion context. The state type is then prefixed with M (match), yielding MP, ML, MR states or with I (insert) yielding IL and IR states. Each state is associated with emission and transition probabilities, which sum to one and are not shown.

State type	Description	Production rules
P	pair emitting	$P \rightarrow xNy$
L	left emitting	$L \rightarrow xN$
R	right emitting	$R \rightarrow Nx$
B	bifurcation	$B \rightarrow SS$
D	delete	$D \rightarrow N$
S	start	$S \rightarrow N$
E	end	$E \rightarrow \epsilon$

While this setup of states and nodes is a de-facto standard, also other model topologies are possible (Janssen and Giegerich, 2015).

Infernal is related to the *HMMER* package that performs similar tasks for *protein* families, which are models by hidden Markov models. Both packages share the *Easel* library and mini-applications that can be used for manipulating sequence data. Following is a description of those tools that were used in the thesis.

cmbuild

cmbuild constructs a covariance model from a multiple sequence with consensus structure, currently only in .stockholm-format. The result are covariance

model complete with nodes, states and transition, as well as emission probabilities. *cmbuild* also computes a *HMM* from the input sequences, which can be used as a pre-filter during homology search and includes it in the output. The result model is not yet ready for direct use in homology search, it first has to be calibrated by **cmcalibrate**.

cmcalibrate

Preparing a model for homology search requires a calibrating step for E-value determination. **cmcalibrate** runs the covariance model against randomly generated sequences and gathers the achieved bit-scores. This process can take hours and can be sped up by reducing the number of random sequences generated, which consequently reduces the number of generated bit-scores. The bit score histogram is then fit to the tail of the exponential distribution. E-values for homology search results can then be estimated with the parameters of this distribution.

cmalign

cmalign is a multiple sequence alignment tool, that aligns a set of input sequences to a covariance model. The resulting stockholm alignment contains the same consensus structure as the one used to construct the covariance model.

cmstat

cmstat computes a set of statistics for an input covariance model, like the number of sequences the model was constructed from, the number of base-pairs and bifurcations. For a calibrated covariance model can also output E-values and bit scores for a given database size.

2.4.2 Rfam - RNA-family database

The **Rfam** database is a steadily growing repository of *RNA*-family models. Currently there are 2474 models (version 12.1) in the database, which represent 9 million annotated *ncRNA* loci.

Each database entry for a family consists of several different elements. Sequences of family members and their genomic coordinates are tabulated together with their organism of origin. A multiple-sequence alignment containing the consensus secondary structure and the covariance model with curation information are the core element of the entry.

Many families are additionally annotated with *RNA* class and *RNA* clan membership. Since **Rfam** version 10 some family entries are linked to corresponding Wikipedia entries.

The database started with 25 families (Version 1.0, 2002-08-15) and has grown on average by approximately by 176 families per year. Since **Rfam** version 6.0 the growth rate has increased and was approximately 237 families per year for for that time interval. The database growth is shown in Figure 2.13.

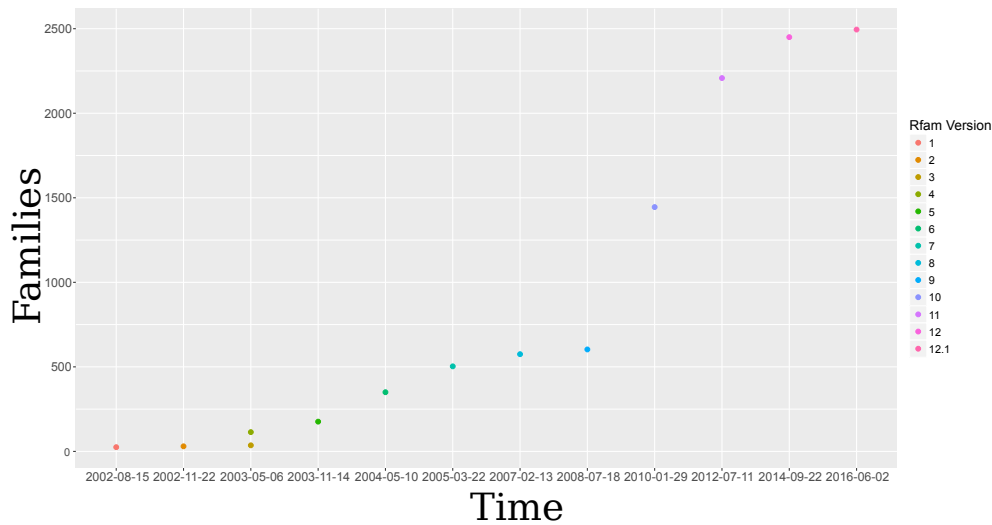


Figure 2.13: **Rfam** database family number development adopted from (Burge et al., 2012). This shows the family number of major **Rfam** releases, with numbers acquired from the family files available via the **Rfam** ftp server. The used dates correspond to the time stamps of the files. The current release, a subversion release (12.1) was also included.

Each family model in the database consists of a **HMMER** hidden Markov model, and of the covariance model. The hidden Markov model serves as pre-filter during homology search and models the sequence of the *RNA*-family. The covariance model represents sequence and secondary structure of the *RNA*-family.

The model construction and curation process is a sophisticated process, that requires an immense computational effort. Moreover not only new families are added to the database also new genomes are constantly sequenced. Existing *RNA* families need to be adjusted to also include scanning results for these families. The process of model construction is explained in further detail in the Chapter 3 and a automatic solution is presented in Chapter 6.

The model construction process depends on homology search tools to identify

additional potential family members.

2.5 Homology search

The search for homolog genes in other species is a sub-field of bioinformatics known as homology search. It utilizes the sequences of the gene in question and contextual information. In case of proteins the frequently conserved codon triplets encoding amino acids can be utilized.

In *RNA* homology search, which this chapter will focus on, the secondary structure, or base-pairing information can be used to trace homolog *RNAs* through evolution.

The same useful abstraction applied to sequences, that was used for sequence alignment can also be applied to homology search. Instead of the whole chemistry of each nucleotide, just the variable component, the nucleobase is considered. We can then use a single letter to represent each nucleotide. Consequently a gene can be understood as a word and a genome as a text. Even with this simplification in mind, genomes represent long texts with millions of characters in case of bacteria, to billions of characters in case of mammals. Some homologs can be easily found by considering sequence information alone, others require the inclusion of secondary structure information, or even orthology.

Matching single genes to genomes is actually a special case of sequence alignment. As presented in the sequence alignment background (see Section 2.2) there are local, semi-global (glocal) and global alignment methods.

The task is to compare a short sequence with a longer one. For homology search local sequence alignment that does not punish unmatched flanking regions with negative scores would be therefore the method of choice.

The downside of local sequence alignment is the required run-time. The Smith-Waterman algorithm has a run-time of $\mathcal{O}(nm)$, where n is the length of the query and m of the sequence database. A current *CPU* performs in the magnitude of 10^9 operations per second. In a very simplified scenario, matching a 100 nucleotide sequence against the human genome with over 10^9 nucleotides would therefore require $100 * 10^9$ operations, which equals 100 seconds.

Lets consider instead a search against a database including a big set of organisms, for example the **RefSeq** database (Pruitt et al., 2012). **RefSeq** release 76 contains sequences from 59995 organisms which total to 825,600,134,816 genomic nucleotides. In this case our search would take hours. Despite the advantage that exact algorithms give all possible solutions, including the best one this is too long for many tasks in bioinformatics.

The requirement for tools that deliver homology search results in a short time has made heuristic approaches very successful in this field. Heuristic algorithms take certain assumptions, which reduce the search space and the necessary run-time. This has the disadvantage that the best solution can be excluded from the possible search results.

There is a range of different implementations available to match a sequence to a genome and many of them are specialized to a specific setting. Three tools for *RNA* homology search will be introduced, **blast** and **nhmmer** in this section and **cmsearch** in the section for *RNA*-family models (see Section 2.3).

The reliable identification of homolog sequences requires not only the tools themselves but also some concepts, from statistics, which will be introduced with **blast**.

2.5.1 BLAST

BLAST (Altschul et al., 1990), or *Basic Local Alignment Search Tool* is a heuristic version of the Smith-Waterman algorithm and with over fifty-thousand citations one of the most highly cited publications in existence.

This high number of citations show the demand for a fast method to search genomic sequences in the last years.

It is a word-based method, that matches a short sub-word and then expands it. The approach consists of three steps.

The seeding step first generates sub-strings from the query sequence. Depending on the bio-polymer type different sub-word lengths are useful. Protein sequences use an initial sub-word size of three, the size of a codon. Nucleotide sequences that lack the reading frame of proteins require longer sub-words, which are per default eleven nucleotides long.

Variants of these sub-words are generated and used to scan the sequence database for exact hits.

The second step is an extension step that expands the sub-words on the query sequence by aligning additional flanking query sequence nucleotides to the exact match. Originally the extension step could not bridge gapped regions, however this functionality has been included (Altschul et al., 1997). The results of this extension are called maximal segment pairs (*MSP*).

Masking can either further speed the **BLAST** search up, by avoiding the matching of specific regions entirely, or increase the sensitivity, by assigning different importance to parts of the query sequence.

BLAST defines hard- and soft-masking. Hard-masking means that regions of

the database are not scanned. Repetitive genomic regions are typically hard-masked. If the *RNA* of interest, e.g. *tRNA* occurs in these regions, it will not be included in the result list.

There are two variants of soft-masking. Database soft-masking only excludes the masked regions in the seeding step, but not in the expansion step. Query soft-masking allows to define conserved regions of the query, which will exclusively be used during the seeding step. The rest of the query sequence can still be used during extension. This type of masking can make searches more profile like and is used in **RNAlien** (see Figures 8.2 and 8.3).

The third step is a evaluation of the maximal sequence pairs for statistical significance. This relies on the E-value the *MSP* achieved. Statistically significant *MSP* hits are called high-scoring segment pairs (*HSP*).

2.5.2 Expected value

The E-value, or *Expected value*, used in sequence similarity search, relies on the assumption that the maximal scores achieved by *MSPs* follow an extreme-value distribution (Karlin and Altschul, 1990; Gumbel, 1958).

E-value is an estimate how many hits in the sequence search would achieve at least that same score by chance. The E-value can be computed from the raw score as follows, the equation is adopted from (Karlin and Altschul, 1990).

$$E = K m n e^{\lambda S} \quad (2.4)$$

Where E is the E-value, S is the raw score, m and n are the query and database length, K is a scaling factor for the search space and λ is a scaling factor for the scoring system.

Alternatively the E-value can be computed via the bit score, which has the unit *bits* (Altschul, 1991, 1993) and can be computed by following equation, adopted from (Altschul et al., 1997):

$$S' = \frac{\lambda S - \ln K}{\ln 2} \quad (2.5)$$

Where S' is the bit score, S is the raw score, K is a scaling factor for the search space and λ is a scaling factor for the scoring system.

The corresponding number of sequences with at least the bit-score S' can be

approximated by the following equation, adopted from (Altschul et al., 1997).

$$E = mn2^{-S'} \quad (2.6)$$

Where S' is the bit score, m and n are the query and database length and E is the E-value.

The E-value can then be used as a cut-off for homology search results.

2.5.3 nhmmer

nhmmer (Wheeler and Eddy, 2013) relies on profile Hidden-Markov models (see Section 2.3) for homology search. Homology search with a hidden Markov model uses the Viterbi algorithm that aligns the genomic sequence against the model.

nhmmer has the advantage over **BLAST** that the used *HMMs* encode which parts of the model are conserved and which not. This increases sensitivity, but has a much higher run-time complexity.

nhmmer uses a set of consecutive pre-filters and applies the Viterbi algorithm only on promising sequence segments, thereby gaining speed via a loss of sensitivity.

2.5.4 cmsearch

cmsearch is a homology search tool that uses calibrated covariance models to search as sequence database. In comparison to **nhmmer**, **cmsearch** also uses secondary structure information.

Scanning a sequence database with a covariance model is an application of the CYK-algorithm. It has run-time complexity of $\mathcal{O}(M_aLD + M_bLD^2)$, where M_a is the number of non-bifurcation states, M_b the number of bifurcation states, L the length of the input sequence database and D the length of the longest aligned subsequence (Durbin et al., 1998). This is achieved by several restrictions, as the run-time of a general purpose CYK algorithm for SCFGs would have a time complexity of $\mathcal{O}(L^3N^3)$ and memory consumption of $\mathcal{O}(L^2N)$, where N is the number of non-terminal states (Durbin et al., 1998).

Moreover **cmsearch** has a set of consecutive heuristic pre-filters, that increase search speed significantly at a loss of sensitivity. **cmsearch** results contain the hit sequences, their sequence database coordinates, as well as bit-score and corresponding E-values.

3 RNA-family model construction

There is a huge potential for adding new *RNA* families to existing databases. The importance of this effort is underlined by the fact that the journal *RNA* biology (Gardner and Bateman, 2009) has set up an own track for publishing novel *RNA*-family models.

This chapter provides the introduction for the publication **RNAlien - Unsupervised *RNA*-family model construction** (see Chapter 6) and provides a description of the state of the art of *RNA*-family construction. Moreover results from a pre-study that was concerned with sequence and structure conservation in **Rfam** families and post-publication results of running **RNAlien** (Eggenhofer et al., 2016) with query-soft-masking are presented.

An **Rfam** *RNA*-family model consists of a structural alignment with the known family members with a corresponding consensus structure and the probabilistic model itself. A covariance model captures the sequence and structure of the family with a stochastic context free grammar representation. This abstraction of the *RNA*-family, can be used to find novel, formerly unknown members of the *RNA*-family via homology search.

3.1 Construction of RNA families and Clans

The building of *RNA*-family models starts with the collection of *RNA* sequences of that family. This can be a number of *RNAs* that has been identified by experiment and cataloged as members of the same family, or alternatively a set of sequences that has been found via homology search.

To maximize the sensitivity of the resulting covariance model in homology search, it is necessary to include a representative sample of family members, in terms of structure and sequence. Biologically relevant *RNA* base pairs are often conserved even after their mutation, to prevent a loss of function. Mutants that have a second mutation in the previously unmutated base pair that restore the base-pair are more prone to survive natural selection processes. The restoration of the original base pair is known as covariance. Ideally the representative first set of member sequences shows this covariance for relevant base pairs.

3.1.1 Seed alignment

Rfam labels this first set as seed sequences. The set of seed sequences needs to be aligned to be able to construct a *RNA*-family model from it. Seed alignments can range from a few sequences to hundreds (e.g. *U4 snRNA* with 140 sequences, *tRNA* with 954 sequences).

3.1.2 Consensus secondary structure

The alignment of the *RNA*-family also includes a consensus secondary structure. This structure is then used in the covariance model.

There are several ways to obtain a consensus secondary structure, The sequence alignment can be performed first and the consensus secondary structure computed afterwards. *RNAalifold* (Hofacker et al., 2002; Bernhart et al., 2008) averages the contributions from secondary structure predictions for the individual sequences (Mathews and Turner, 2006; Lorenz et al., 2011), according to the input alignment. These energies are then used to solve the folding problem.

The alternative are structural alignment algorithms (see Subsection 2.2.2), which use the secondary structure during the alignment process and also yield a consensus secondary structure.

3.1.3 Alignment to model

However the curation process goes beyond selecting representative sequences for the seed alignment. The seed alignment is also used to construct the covariance model of the family.

Tools from the **Infernal** package are using the alignment with the consensus secondary structure to build the covariance model (see Subsection 2.4.1) **cmbuild** constructs the actual model, while **cmcalibrate** calibrates the parameters for E-value estimation.

This model is then used to find all instances of the *RNA* in the sequence database of **Rfam** and select candidates for the full alignment.

To determine which sequences will be accepted into the full alignment there are three different cutoffs defined by **Rfam**. These cutoffs refer to bit score cutoffs resulting from searching the sequence database with **cmsearch**.

The *noise cutoff* is the bit-score that the best hit received that is defined as noise by the curator.

The *gathering cutoff* is the bit score that hits have to achieve to be included in the full alignment.

The *trusted cutoff* is the lowest bit score a sequence yielded that is considered a true family member.

These cutoffs were introduced before the computation of E-values was implemented in the **Infernal** toolkit. Possibly these bit-score cutoffs will be changed to E-value thresholds in the future.

3.1.4 RNA-family clans

For some families the process of selecting sequences and setting thresholds can lead to problematic results. This lead to the introduction of *RNA*-family clans (Gardner et al., 2011).

Families that share the same ancestor and biological function but are difficult to align, or can be aligned and have different biological functions can be grouped into clans.

The *RNAase P* family has been split into three families, each for a different domain in the tree of life. The family as a whole is too diverse for good quality alignments.

3.1.5 Full alignment

The model can be used to search genomes for more members of this *RNA*-family. From the perspective of a single *RNA*-family that means it is possible to find paralogs in one species and orthologs in other species. The sequences obtain with the homology search by `cmsearch` that satisfy the gathering cutoff are included in the full alignment.

Full alignments range from a two sequences for a family up to 3,302,554 sequences for the bacterial small ribosomal sub-unit. The number of sequences in `Rfam` seed and full alignment is shown in the following Figure 3.1.

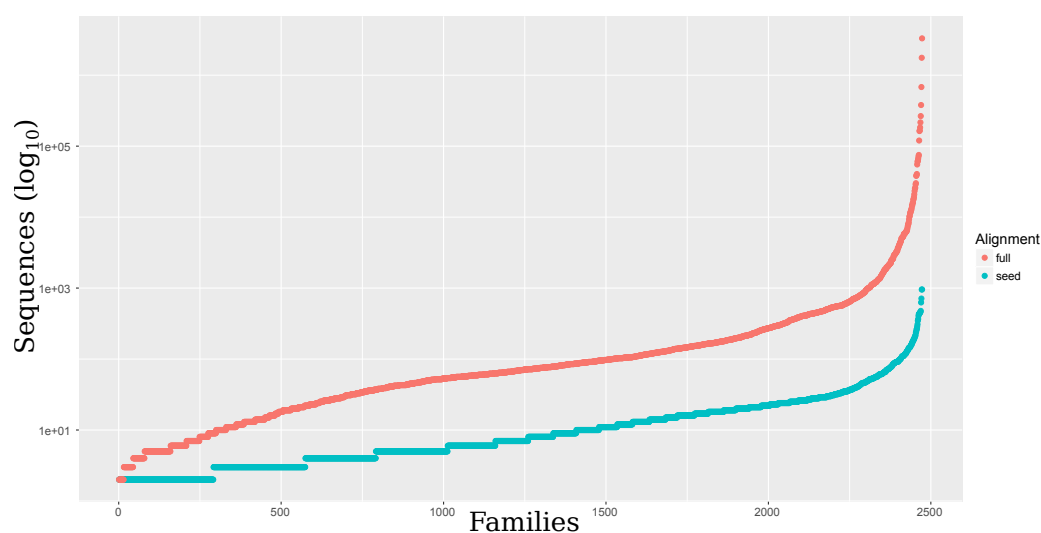


Figure 3.1: Sequences per family `Rfam` seed and full alignments: Full alignments are shown in red, seed alignments in blue. The y-axis is \log_{10} transformed.

3.2 Conservation of Rfam families

RNA families share a common biological function in different organisms. These *RNA* molecules rely on their specific structure to fulfill their function. The sequences used to build *RNA*-family models are therefore expected to exhibit structure conservation.

The Rfam (Nawrocki et al., 2014b) seed alignments were investigated regarding their structuredness, as a pre-study for *RNAlien* (Eggenhofer et al., 2016). The goal was to determine if there is an overall conservation of secondary structure or if it is limited to subgroups or only individual families.

An established measure for structure conservation, the structure conservation index (*SCI*) (Washietl et al., 2005; Gruber et al., 2010) was computed for all 2450 of Rfam 12.0 seed alignments. The results for all families and the available subgroups are shown in 3.1 and 3.2.

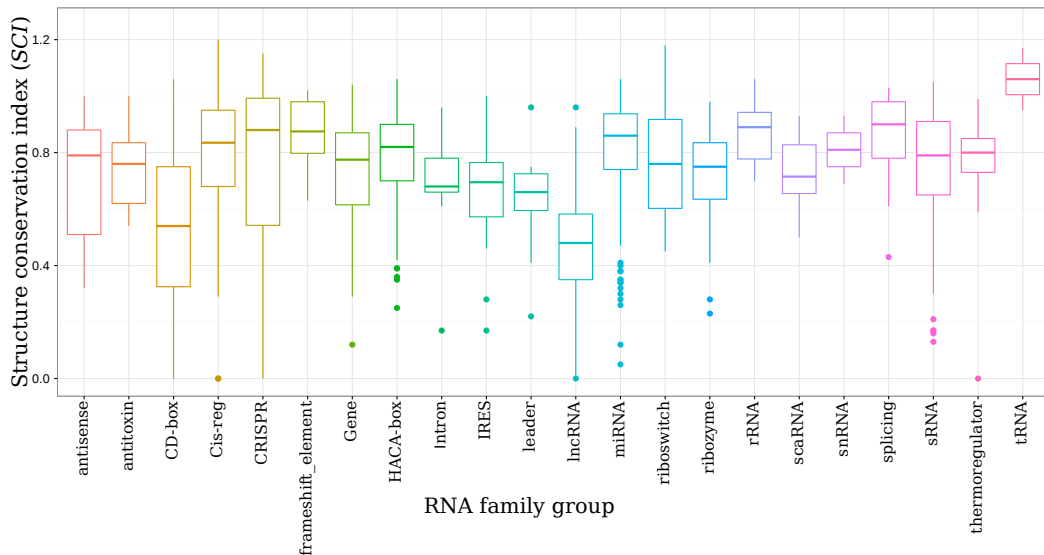


Figure 3.2: Structure conservation index (*SCI*) of Rfam family groups: The family groups show in general structuredness, with the exception of the *lncRNA* group. The *SCIs* for families in one group can be spread, this can be observed particularly in *CRISPR*, *antisense* and *CD-box* subgroups.

The results show clearly that both the subgroups and the Rfam seed alignments show structure conservation, with the one exception of long non-coding *RNAs*. This lack of conserved secondary structure in long non-coding *RNA*, has also been confirmed by detailed investigation (Rivas et al., 2016).

However a high structure conservation is to be expected for *RNAs* that have a high level of sequence conservation, because base-pairs are conserved as well.

Table 3.1: Structure conservation index *SCI* for *Rfam* subsets: The seed alignments have a average *SCI* of approximately 0.7, which indicates structure conservation. The *frameshift_element* family group has the highest average *SCI*, while *lncRNAs* have the lowest average structure conservation.

Rfam subset	Min	Max	Median	Mean
All	0.00	1.20	0.76	0.70
Cis-reg	0.00	1.20	0.80	0.78
frameshift_element	0.63	1.02	0.87	0.85
IRES	0.17	1.00	0.70	0.68
leader	0.22	0.96	0.68	0.66
riboswitch	0.30	1.18	0.77	0.76
thermoregulator	0.00	0.99	0.80	0.71
Gene	0.00	0.99	0.80	0.71
antisense	0.05	1.02	0.51	0.54
antitoxin	0.54	1.00	0.76	0.74
CRISPR	0.00	1.15	0.88	0.72
lncRNA	0.00	0.96	0.48	0.47
miRNA	0.05	1.06	0.86	0.81
ribozyme	0.17	0.98	0.71	0.69
rRNA	0.08	1.06	0.68	0.64
snRNA	0.00	1.06	0.69	0.63
snoRNA	0.00	1.06	0.68	0.63
CD-box	0.00	1.06	0.54	0.53
HACA-box	0.25	1.06	0.82	0.79
scaRNA	0.50	0.93	0.71	0.72
splicing	0.17	1.03	0.80	0.76
sRNA	0.13	1.05	0.80	0.76
tRNA	0.17	1.17	0.71	0.76
Intron	0.17	0.96	0.68	0.68

Covariant base-pairs are a strong indicator for functional conservation, but we can only observe them if the sequence is not conserved. The next goal was therefore to investigate the sequence conservation of the *Rfam* seed alignments. The mean sequence identity was computed for all *Rfam* seed alignments with *RNAz* and is shown in Table 3.2 and in Figure 3.3.

The structure conservation index is on average lower than the mean sequence identity for all *Rfam* seed alignments. However for *tRNA*, *miRNA*, *splicing*, *intron*, *riboswitch* and *ribozyme* subgroups the *SCI* is higher than the *MSI*. While the human curator is able to consider different criteria, this ratio of sequence and structure conservation presents a very conservative measure for automatic family construction. This concept is used in *RNAlien* (Eggenhofer

Table 3.2: Mean sequence identity (MSI) for **Rfam** family groups: The mean sequence identity on average is lower than the structure conservation index. The family group with the lowest average MSI is the Intron group and the one with the highest the CRISPR group.

Rfam subset	Min	Max	Median	Mean
All	0.24	1.00	0.80	0.79
Cis-reg	0.43	1.00	0.80	0.79
frameshift_element	0.62	0.95	0.86	0.86
IRES	0.55	1.00	0.83	0.81
leader	0.43	0.97	0.71	0.71
riboswitch	0.43	0.79	0.68	0.66
thermoregulator	0.55	0.93	0.84	0.80
Gene	0.55	0.93	0.84	0.80
antisense	0.54	0.95	0.80	0.79
antitoxin	0.66	0.88	0.81	0.79
CRISPR	0.66	1.00	0.89	0.87
lncRNA	0.60	0.96	0.80	0.79
miRNA	0.48	1.00	0.80	0.79
ribozyme	0.24	0.97	0.69	0.68
rRNA	0.24	0.87	0.78	0.75
snRNA	0.34	1.00	0.80	0.79
snoRNA	0.34	1.00	0.80	0.79
CD-box	0.34	1.00	0.80	0.79
HACA-box	0.48	1.00	0.80	0.79
scaRNA	0.71	0.87	0.81	0.80
splicing	0.24	0.97	0.77	0.75
sRNA	0.27	1.00	0.81	0.81
tRNA	0.24	0.88	0.70	0.69
Intron	0.24	0.73	0.59	0.59

et al., 2016) as *normalised structure conservation index*.

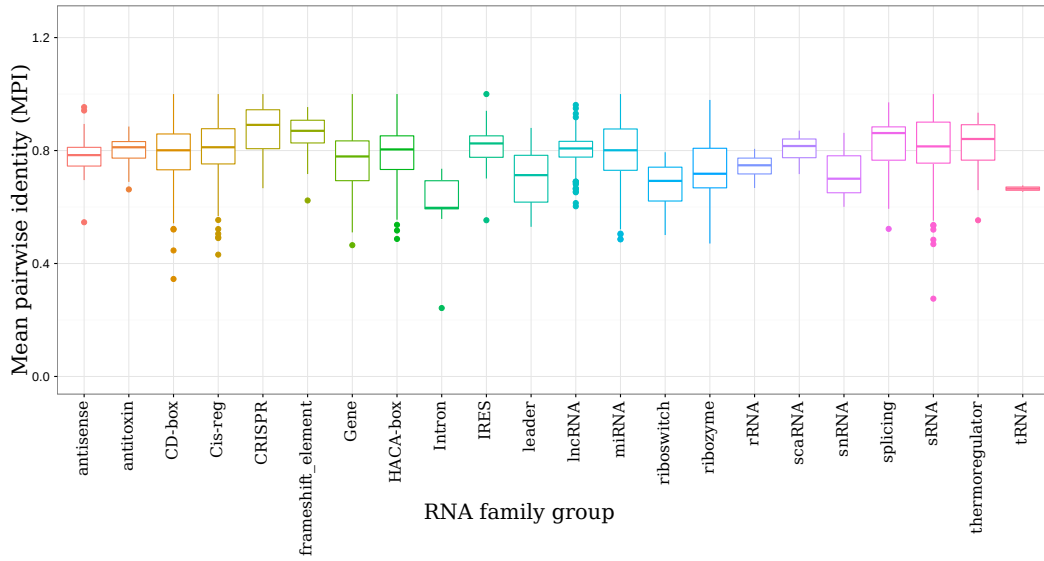


Figure 3.3: [Mean sequence identity (*MSI*) for *Rfam* family groups: The box-plot shows that the *MSI* is on average higher than the *SCI*. The *MSI* values show much less fluctuation within the groups compared to the *SCI*.



Figure 3.4: Mean sequence identity (*MSI*) vs structure conservation index (*SCI*) of *Rfam* family subsets: The (*SCI* and (*MSI*) for each *Rfam* seed alignment is shown as a dot in the plot. Subgroups are labeled in the same color. There is a general trend for families with higher sequence conservation to also have high structure conservation, which can be observed by the bulk of the points being in the right upper quadrant.

4 RNA family model evaluation

A newly constructed *RNA*-family model should be evaluated before further use in homology search, or other applications. In this regard the model itself, the input sequences, the host organisms and the homology search results of the model can be investigated. In this chapter, and the publication in the following Chapter 7, the focus will be on evaluating the model.

The necessity for evaluation models will further grow in the future with the increase of the number of known models in the databases.

The model and its homology search performance depend on the sequences used to construct it. In general it is useful to include all known family members into the construction and especially the most divergent ones.

RNA-family models built from sequences that are not representative for the family lead to a lack of sensitivity in homology search. **RNAlien** provides **RNAz** statistics that include the mean sequence similarity, which enables to identify such sequence sets.

However the selection of too divergent family members can be problematic as well. Specificity of homology search results can be decreased for families that are very divergent in terms of sequence and structure, or families which have closely related families.

Low specificity can be discovered by visualization of search results (See Chapter 5) or by directly comparing the resulting covariance models.

4.1 CMCompare

Specificity in context of *RNA* homology search means, that a specific *RNA* sequence should only be detected by the *RNA*-family model representing this specific *RNA*-family. If an other *RNA*-family model identifies the *RNA* sequence as a significant hit, there is a lack of specificity in one or both models. The optimal alignment of a sequence to a *RNA*-family model is computed with the CYK algorithm. A sequence achieving a high CYK score in the *RNA*-family models of two different *RNA* families would indicate such a lack of specificity.

The **CMCompare** (Höner zu Siederdisen and Hofacker, 2010) algorithm identifies the shared sequence with the highest CYK score in both models. This is accomplished via a tree alignment approach for the guide tree of both models and the associated sequence information.

The minimum of the CYK scores, of this sequence, in the first covariance model c_1 and the second model c_2 is called *link score* and is computed as following (Höner zu Siederdisen and Hofacker, 2010):

$$\text{Linkscore}(c_1, c_2) = \max\{\min\{\text{CYK}(c_1, s), \text{CYK}(c_2, s)\} | s \in \mathcal{A}^*\} \quad (4.1)$$

where \mathcal{A} is the alphabet of nucleotides, \mathcal{A}^* are all strings over \mathcal{A} and s is the shared input sequence between both model, element of \mathcal{A}^* . A straightforward implementation of this equation would require enumeration of all possible sequences yielded by combining nucleotides from \mathcal{A}^* .

CMCompare (Höner zu Siederdisen and Hofacker, 2010) provides a implementation of the **CMCompare** algorithm. This implementation uses a dynamic programming approach that reformulates above equation as recursion. The recursion essentially aligns the guide trees of both covariance models with each other. The time complexity of this approach is $\mathcal{O}(n_1 n_2 l^2)$, where n_x is the number of states in the guide tree of model x and l the number of children per state.

CMCompare uses the a similar scoring scheme as the tools of the **Infernal** (Nawrocki et al., 2009; Nawrocki and Eddy, 2013) package. Search results with *bit scores* over 20 would be considered as a indicator for low specificity. High *link scores* can occur between *RNA*-family models that are part of the same *RNA*-family clan.

Besides the *link sequence* and the *link score*, **CMCompare** also returns the index intervals of the nodes the *link sequence* was emitted from. The regions of both models that are linked can be identified in this fashion.

Curators that are familiar with the biological function of different regions of a *RNA* can use this information to interpret if a high *link score* originates from a homology or from a specificity problem.

The **CMCompare webserver: comparing RNA families via covariance models** (Eggenhofer et al., 2013) (see Chapter 7), provides a web based interface to evaluate covariance models with **CMcompare**.

5 RNA-family member visualization

Once a *RNA*-family model is constructed it can be used to detect novel members of this family. The **Infernal** suite offers the `cmsearch` tool specifically for this purpose.

The found potentially homolog *RNAs* can be considered in its genomic context from a single organism point of view. However the construction of these hubs is a tedious task. Therefore I contributed two tools that provide a simple and efficient way to generate trackhubs and assembly hubs to the **ViennaNGS** toolkit (Wolfinger et al., 2015). The section Track and Assembly Hub construction presents tools for such a purpose.

Alternatively the taxonomic or phylogenetic perspective can be relevant for the research question. Taxonomy tools, as described in the section RNA family members and taxonomy, can be used to visualize the host organisms of the *RNA* in the tree of life.

5.1 Genome browser visualization

Visualising genes in their genomic context can be accomplished by the use of genome browsers. They display the genome as an graphical representation, e.g. a line. Genes and other features, like repeats, are annotated with icons that are located on the genome.

The genome can be indexed via its nucleotide number, called coordinates in the genome browser context. While the whole genome usually has an overwhelming number of features, the start and end coordinates known for each individual feature allow to consider only the features for a specific region.

The genome can therefore be shown in different level of detail, which is extremely useful for the putative *RNA*-family members found during *RNAlien* model construction, or predicted with the result covariance model via *cmsearch*. On the gene level it is possible to verify if the locus overlaps with other annotated genes. This could be a indicator for specificity overlaps as discussed in Chapter 7.

Due to recombination events it is more likely that neighboring genes are involved in a common biological function or process. Investigation of the functional annotation of neighboring genes could therefore be informative in the identification of the biological role of the *RNA* family of interest.

The region in which the gene of interest is embedded can be inspected for the presence or enrichment of specific features, like repetitive regions. This can be useful context information for non-coding *RNAs* which are known to be found frequently in such regions.

Genome and the features are the most basic information provided by genome browsers. The available genome browsers can be distinguished by the additional information they can show in conjunction with the genome.

A driving force for the development of genome browsers and the inclusion of new visualisation has been sequencing in general and next-generation sequencing (Goodwin et al., 2016) (*NGS*). *NGS* is a constant source of new context data that can be intersected with genomic features, like those obtained from *RNAlien*.

Context data like the level of expression for a certain genomic region (Steijger et al., 2013), or its degree of methylation (Bock, 2012) can be annotated in the genome browser as so called tracks. Multiple tracks can align different context features simultaneously to a genomic region.

The versatility of tracks and the availability of ready-to-use context informa-

tion sets genome browsers apart from each other. The **UCSC genome browser** (Kent et al., 2002), offers a extensive (Rosenbloom et al., 2015) number of various tracks for model organisms.

It is possible to add own tracks via track-hub (Raney et al., 2014) to these existing genome browser instances. Moreover new genome browser instances, for organisms not originally included into the **UCSC genome browser** can be included with assembly hubs.

Tools to automatically construct track-hubs and assembly hubs are introduced below and were contributed by the author to **ViennaNGS** (Wolfinger et al., 2015).

5.1.1 Trackhub construction

Trackhubs can be used to include additional tracks into the **UCSC genome browser** instance for a available organisms. These available organisms cover eukaryotic model organisms, specifically from the kingdom of *Animalia* and *viruses*.

Homo sapiens has the biggest number of additional tracks in nine different categories (Mapping Sequencing, Genes and Gene Predictions, Phenotype and Literature, mRNA and EST, Expression, Regulation, Comparative Genomics, Variation, Repeats). The other organisms have subsets of these available.

Discrete track data, like genes can be encoded as **browser extensible format** (BED). The minimal definition necessary for a gene is the identifier of the host *DNA* molecule and the start and end coordinate of the feature.

BED format can be optionally extended to contain a label, a weight (score), strand information, color and blocks. These can be used to include information specific for homology search. The label can be set to the name of the predicted *RNA*, the score can be weighted by the achieved e-value. Color can be used to distinguish hits for different types of predicted *RNAs* and blocks can be used to represent exons and introns.

RNA homology search using a covariance model is usually performed with **cmsearch**. **RNAlien** contains the **cmsearchToBed** tool allowing the conversion of potential *RNA* gene loci from **cmsearch** to BED format with a specifiable E-value or bit-score cutoff.

Continuous track data can encoded in wiggle (**WIG**) format, which is useful to represent e.g. expression strength or the methylation level of genomic regions. Both BED and WIG information can be directly integrated into trackhubs. However data-sets can become large enough to negatively affect the performance

of the browser visualisation engine. Therefore typically data compressed versions of BED and WIG, namely BIGBED and BIGWIG are used. `cmsearchToBed` optionally enables automatic sorting of the BED entries, by coordinates, which is the prerequisite for compression into BIGBED format.

The `track_hub_constructor` tool, available via **ViennaNGS** (Wolfinger et al., 2015), builds the required trackhub data-structure. Usage of this tool and the required parameters are demonstrated using homology search results for splicosomal *RNAs*.

The benchmark data-set for **RNAlien** included three splicosomal *RNA* families, *U6 snRNA*, already presented in the introductions, as well as *U1* and *U2 snRNAs*. For each of these families a covariance model was constructed.

These covariance models were used to search chromosome 16 of the human genome, with Sequencing/Assembly provider ID *GRCh38 Genome Reference Consortium Human Reference 38 (hg38)*. The resulting homology search `cmsearch` results were converted into BED files with `cmsearchToBed` and compressed into BigBed files with *UCSC bedToBigBed* program.

Since the **UCSC genome browser** imports the trackhub via a publicly accessible unified resource locator (URL), this needs to be hard-coded into the trackhub. The trackhub data-structure has to be deposited at that location. The same applies to the generated BIGBED and optionally, and not included in this example, BIGWIG files.

The parameters required by `track_hub_constructor` are the UCSC genome identifier, which enables to map the trackhub to the correct genome, a name for the track hub which will be displayed in the browser (see Figure 5.1). Furthermore the local output file-path and the URL where the data-structure will be available, as well as the URLs of BIGBED and BIGWIG have to be provided. The current example was constructed with the parameters shown in Table 5.1.

The trackhub is still available and can be imported and viewed by navigating the browser to <https://genome.ucsc.edu>. Select *My Data* in the top navigation bar and select the *trackhubs* sub-menu. The browser is redirected to the Track data hubs page, showing all publicly available hubs that can be enabled for genome browser instance. Select the second tab *My hubs* and paste the URL to the constructed trackhub, which is <http://www.bioinf.uni-freiburg.de/~egg/snRNA/hub.txt> for the example case. The browser is then automatically redirected to the correct genome browser instance with the additional trackhub.

Table 5.1: track_hub_constructor parameters with example values as used for building the trackhub shown in Figure 5.1. The hashtag characters are used to delimit the *URLs* on the command line.

track_hub_constructor		
<i>parameter</i>	<i>description</i>	<i>value</i>
-gi	UCSC identifier	hg38
-name	Trackhub name	snRNA
-out	Output path	/home/user/public.html/
-baseurl	URL for UCSC import	http://www.bioinf.uni-freiburg.de/~egg/snRNA
-bigbeds	List of bigbed tracks	http://www.bioinf.uni-freiburg.de/~egg/u6.bigbed #http://www.bioinf.uni-freiburg.de/~egg/u1.bigbed #http://www.bioinf.uni-freiburg.de/~egg/u2.bigbed
-bigwigs	List of bigwig tracks	

Upon loading the example trackhub for *snRNAs* in the *hg38* instance of the genome browser, the *U1*, *U2* and *U6* tracks are inserted and a corresponding control panel below is added (see Figure 5.1). The homology search hits can now be compared to the existing annotation, as all features are aligned to the genomic coordinate ruler shown at the top.

The inserted tracks are shown directly below the genomic ruler, and show a predicted *U6 snRNA* locus. Gencode 22 (Harrow et al., 2006) annotates a *U6 RNA* at this locus that overlaps with the predicted hit from the *RNAlien* model. Moreover we can see that this region is not very conserved over 100 vertebrate species, but strongly conserved in *Macaca mulatta* and partially conserved in *Mus musculus*. The track for repetitive regions shows the *U6 RNA* locus is bordering to a short interspersed element (*SINE*) region.

Further investigation of a novel *RNA* could consider conserved regions in other organisms and overlapping genomic features. For the *U6 RNA* example, that would mean investigating the conserved loci in *Macaca mulatta* and *Mus musculus*. If the novel *RNA* is generally co-located with *SINEs*, or even derived from them, should also be investigated. If such a co-location is present, this simplifies the identification of additional family members.

5.1.2 Assembly hub construction

Genome browser instances are not available for all species as mentioned above. While some specialized sister projects of the *UCSC* genome browser exist, that offer e.g. instances for bacterial species, many organisms are not covered. However it is possible to construct own assemblyhubs that can be imported in

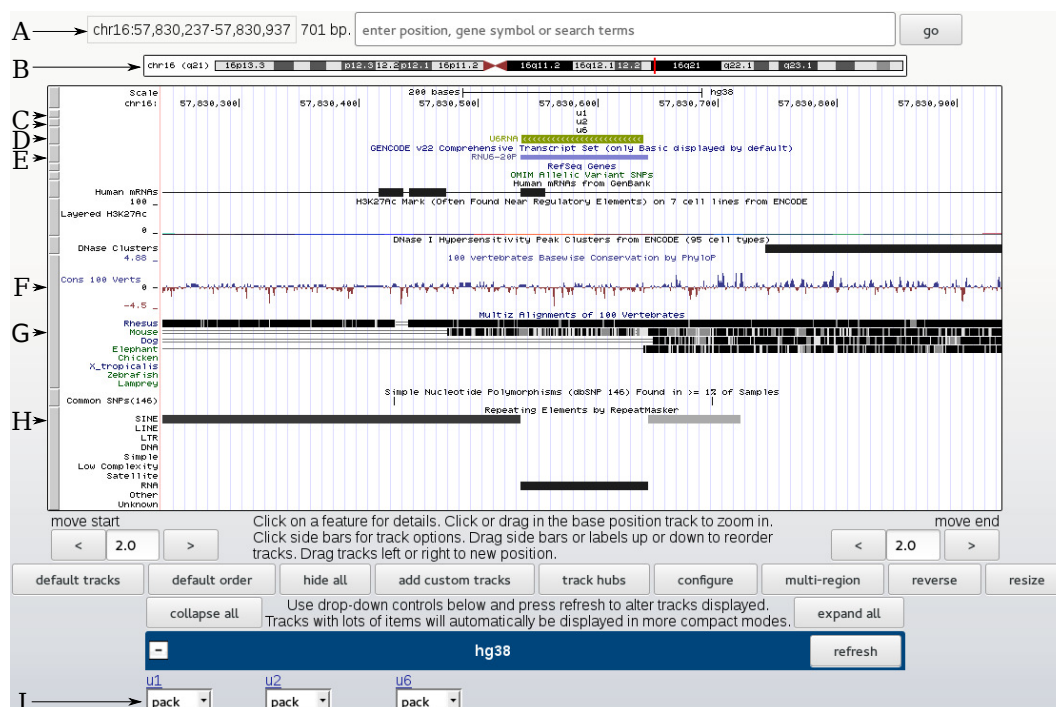


Figure 5.1: Trackhub for predicted splicosomal *RNAs* in *Homo sapiens* This figure shows the UCSC genome browser instance extended with a trackhub for *U1, U2, U6* splicosomal *RNAs*. The *RNAs* in the trackhub were annotated with covariance models constructed by *RNAlien*. At the top the displayed genomic coordinates are shown (A). This region is also marked in the graphical depiction of the chromosome with a red vertical bar (B). The box that dominates the center of the figure shows the tracks vertically stacked on each other and aligned to the genomic coordinate ruler shown at the top of the box. The *U1 snRNA* and the *U2 snRNA* tracks (C) do not have genes predicted in this region. However the *U6 RNA* track features a green box representing an annotated gene (D). Moreover this gene is nearly identical with the annotation from GENCODE 22 Harrow et al. (2006) for this *U6 RNA* loci (E). The following tracks following below contain context information, like the overall conservation of this region for 100 vertebrate species (F), the individual conservation for eight model organisms (G) and different types of repetitive regions (H). At the bottom of the figure control elements for the trackhub (*hg38*) are shown(I).

the *UCSC* genome browser.

The *assembly_hub_constructor* tool also included in *ViennaNGS* (Wolfinger et al., 2015) can build assembly hub data structures. The usage will be demonstrated using the *tRNA* and *tmRNA* covariance models from the benchmark results of *RNAlien* applied to *Escherichia coli K-12*.

The organisms genome is needed by `assembly_hub_constructor` to build the index to which all genomic features will be aligned to. Moreover there is no automatically available annotation present which can be used to put the newly added tracks into context.

Nevertheless annotation data available from additional data-sources can be used and included together with the homology search tracks. Genomes available via NCBI genbank (Benson et al., 2008) have such annotation data encoded in genbank format (gbk). The `gff2bed` tool, also part of the **ViennaNGS** suite, can partition genomic features according to type and convert them into **BED** format. In this way tracks for coding sequences, non-coding *RNAs*, repetitive elements and others can be obtained in a single step.

The local directory path to the **BED** files for annotation, those from homology search and **WIG** files is required by `assembly_hub_constructor`. All files will be automatically converted to the corresponding compressed format and included into the assembly hub.

The example assembly hub was constructed with the parameters listed in Table 5.2. The resulting genome hub can be used by providing the URL <http://www.bioinf.uni-freiburg.de/~egg/assemblyHub/hub.txt> to the UCSC **genome browser**, as described for the trackhub.

Table 5.2: `assembly_hub_constructor` parameters with example values as used for building the trackhub shown in Figure 5.2

assembly_hub_constructor		
<i>parameter</i>	<i>description</i>	<i>value</i>
<code>-fasta</code>	Fasta filename	GCA_000005845.2_ASM584v2_genomic.fna
<code>-name</code>	Assembly hub name	U00096.3
<code>-in</code>	Directory with bed and wig files	bed
<code>-out</code>	Output path	/home/user/public.html/
<code>-baseurl</code>	URL for UCSC import	http://www.bioinf.uni-freiburg.de/~egg/assemblyhub

The resulting genome browser instance, in Figure 5.2, shows the genomic ruler for the provided *Escherichia coli* U00096.3 on top. The annotation data imported from NCBI genbank (Benson et al., 2008) consists of a coding sequence (*CDS*), gene, miscellaneous features, mobile elements, replication origin, ribosomal *RNA*, *tmRNA* and *tRNA* tracks. Tracks generated from the **cmsearch** homology search results are shown at the bottom.

All annotated *tRNAs* are overlapping with loci predicted by the **RNAlien** model, which predicts an additional *tRNA*. Depending of the annotation quality of the organism in question, this could be a homolog gene, an inactive

variant of the gene (e.g. pseudogene) or a spurious hit. In case of the *tmRNA*, neither the NCBI annotation nor the homology search have features in this region.

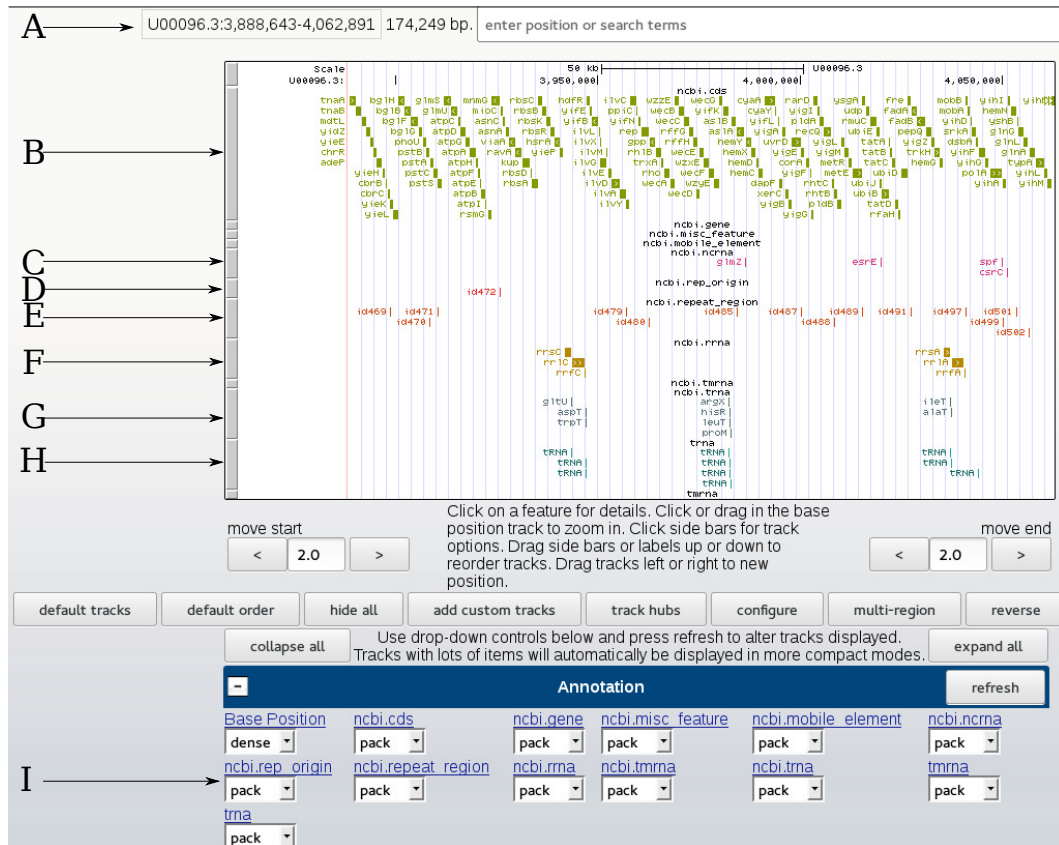


Figure 5.2: Assemblyhub for predicted *tRNAs* in *Escherichia coli* This figure shows the a novel UCSC genome browser instance with annotation tracks generated from NCBI genbank (Benson et al., 2008). The *tRNAs* in the trackhub were annotated with covariance models constructed by RNAlien. At the top the displayed genomic coordinates are shown (A). The box that dominates the center of the figure shows the tracks vertical stacked on each other and aligned to the genomic coordinate ruler shown at the top of the box. NCBI genbank annotations with features in this region are coding sequences(B), non-coding *RNAs* (C), replication origin(D), repeat regions(E), ribosomal *RNAs* (F) and *RNAs* (G). *tRNA* features identified by the RNAlien (H) model are overlapping with all genes in the NCBI annotation. At the bottom of the figure control elements for the tracks shown(I).

5.2 RNA-family members and taxonomy

RNA-family members exist in different host organisms. The number of organisms and the number of family members in these organisms, can be used to evaluate how widespread and variable the family is. However this does not take the diversity of the host organisms into account.

The taxonomy of the hosts contains this diversity information and allows to map the presence of the *RNA*-family to the tree of life. The distribution of the family members in this tree can give clues about the evolutionary history of the family or the possible function of the *RNA* in context with other information. *RNA* families can be restricted to only a few closely related species or spread over multiple kingdoms or even the whole tree. Disconnected sub-trees featuring the *RNA*-family could be interpreted as either gene transfer to some species or loss of the gene in others.

While there is a huge number of different tools to visualize phylogenetic trees, the selection for taxonomic trees is small. Interactive tree of life (*iTo1*) (Letunic and Bork, 2016) is the most popular webservice in this regard and offers a wide range of functionality. However it is only available online and not as a downloadable tool.

The following text describes two tools from the package **Taxonomy Tools**, which has been developed to visualize, process and compare taxonomic trees. **Taxonomy Tools** has been written in the **Haskell** programming language. It is publicly available via **GitHub** (<https://github.com/eggzilla/TaxonomyTools>).

5.2.1 Visualizing taxonomy of *RNA*-family members

In order to visualize the taxonomy of all *RNA*-family members, the member sequences need to be associated with their host organisms. NCBI taxonomy (Federhen, 2012) provides tables that associate organisms with unique taxonomic ids, rank, genetic code and literature references. Obtaining the taxonomy id for a specific sequence depends on its available context information.

The accession number that identifies the *DNA* molecule or a gene identifier (*gi*) can be used to find the corresponding organism. **Taxonomy tools** contains the **Accessions2TaxIds** helper tool to automatically perform this conversion step. **RNAlien** (Eggenhofer et al., 2016) stores and outputs the taxonomy ids, for organisms containing *RNA*-family members, at the end of a construction process.

`TaxIds2Tree` uses the following steps to construct and visualize taxonomic trees. For each taxonomy id the more general parent taxonomic node as described in Subsection 2.1.8 is tabulated. These tables are parsed via the `parsec` Leijen (2001) library.

The list of organism identifiers is used to recursively obtain the identifiers of all parent nodes up to the root from the parsed tables. The set of node id and parent node id tuples is representing the edges of the taxonomy tree. The unique list of the taxonomy identifiers from the tuple represents all our included taxonomic nodes.

`Taxonomy tools` uses these edges and nodes and a general datastructure from the functional graph library (Erwig, 2001) to represent taxonomic trees.

`TaxIds2Tree` requires the list of organism taxonomy ids and allows to optionally toggle the rank of each taxonomic node.

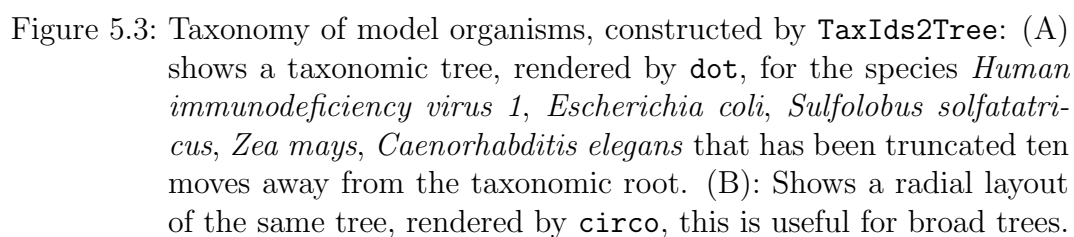
The result trees are optionally either encoded in `.dot` or in `.json` format. `.dot` format can be used as input for the `Graphviz` (Gansner and North, 2000) toolkit. `Graphviz` implements a series of different lay-outing algorithms for graphs. The `dot` tool provides hierarchical drawings of graphs and provides both vertical (see Figure 5.3 A) and horizontal (see Figure 5.5) layouts for trees. Among others also radial layouts with the `circo` tool are possible (see Figure 5.3 B). This versatility is well suited to produce a non-overlapping and clear rendering of the tree.

The resulting figure can become very broad in one dimension if organisms are from divergent lineage. Therefore it is possible to control the depth of the tree, by defining the edge distance from the root node.

Alternatively the `.json` format output can be used with the data-driven documents (Michael Bostock, 2011) (`D3js`) javascript library. This enables interactive taxonomic trees that can be embedded in webpages. These trees are scroll- and zoomable, which enables the rendering of very large trees. Additionally it is possible to collapse sub-trees by clicking on the corresponding node, which allow to exclusively display the relevant parts of the tree. This was used for the `RNAlien` webservice that has been published alongside the tool to visualise the taxonomy of detected *RNA*-family members (see Figure 5.4).

5.2.2 Comparing taxonomy of *RNA*-family members

The distribution of traits over different species is a relevant question for classical phylogeny, while the distribution of homolog genes is interesting for molecular biology.



These distinct features can each be mapped individually to the taxonomic

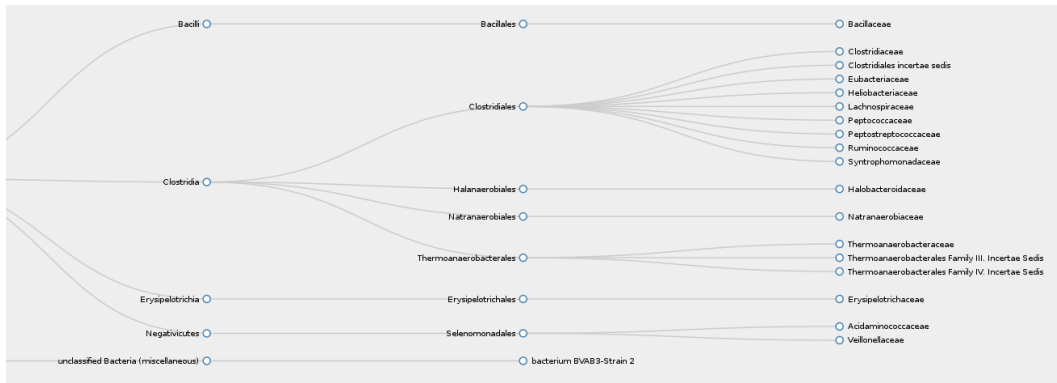


Figure 5.4: Interactive taxonomic tree, constructed by `TaxIds2Tree` and rendered by `3Djs`: the tree is zoom- and scrollable, sub-trees can be collapses by clicking.

tree as described above. However traits as well as genes can be considered in context to each other.

Taxonomy Tools contains the `TaxIds2TreeCompare` tool which accepts sets of taxonomy identifiers. Each of these sets represents a group that shares a certain feature, for example the presence of a specific *RNA* family. The sets in combination represent the distribution and overlap of the features over several organisms. `TaxIds2TreeCompare` visualises these features in the taxonomic tree.

For each set a color is selected from the spectrum. Each provided leaf node is labeled with the colors corresponding to the features it has been assigned. `TaxIds2TreeCompare` intersects the features present in child nodes for each internal taxonomic node. This enables to see at which point features overlap. A example for this in horizontal tree layout shows a actual example for *CRISPR-Cas* (Maier et al., 2016; Makarova et al., 2015) system in *Haloarchaea* (see Figure 5.5).

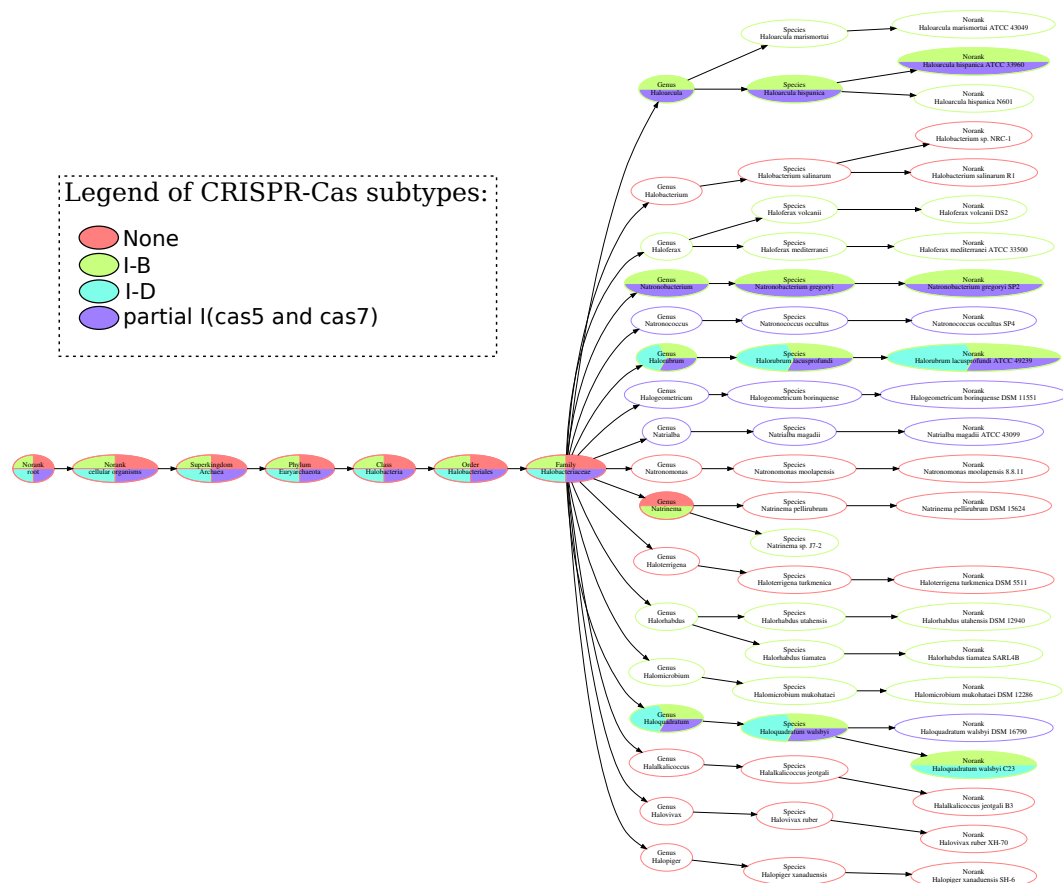


Figure 5.5: Comparing taxonomy distribution of *CRISPR-Cas* (Maier et al., 2016; Makarova et al., 2015) systems in *Haloarchaea*. This figure was automatically build with *TaxIds2TreeCompare* and *dot* and shows the distribution of three different sets of Cas-protein in *Haloarchaea*. The legend was added manually.

6 Paper: RNAlieN - Unsupervised RNA family model construction

Florian Eggenhofer, Ivo L. Hofacker, Christian Höner zu Siederdisen

RNAlieN - Unsupervised *RNA* family model construction

Nucl. Acids Res. - published/publiziert - 22.06.2016

Florian Eggenhofer created the programs and the webservice. All three authors participated in writing the paper.

Bibliography:

Florian Eggenhofer, Ivo L. Hofacker, and Christian Höner zu Siederdisen. RNAlieN Unsupervised RNA family model construction, Nucl. Acids Res. first published online June 21, 2016 doi:10.1093/nar/gkw558

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

RNAlien – Unsupervised RNA family model construction

Florian Eggenhofer^{1,2,*}, Ivo L. Hofacker^{1,3} and Christian Höner zu Siederdissen^{1,4,5}

¹Institute for Theoretical Chemistry, University of Vienna, Währingerstrasse 17, A-1090 Vienna, Austria,

²Bioinformatics Group, Department of Computer Science University of Freiburg, Georges-Köhler-Allee, 79110

Freiburg, Germany, ³Research Group Bioinformatics and Computational Biology, Faculty of Computer Science, University of Vienna, A-1090 Vienna, Austria, ⁴Bioinformatics Group, Department of Computer Science, University of Leipzig, D-04107 Leipzig, Germany and ⁵Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany

Received October 31, 2015; Revised June 06, 2016; Accepted June 08, 2016

ABSTRACT

Determining the function of a non-coding RNA requires costly and time-consuming wet-lab experiments. For this reason, computational methods which ascertain the homology of a sequence and thereby deduce functionality and family membership are often exploited. In this fashion, newly sequenced genomes can be annotated in a completely computational way. Covariance models are commonly used to assign novel RNA sequences to a known RNA family. However, to construct such models several examples of the family have to be already known. Moreover, model building is the work of experts who manually edit the necessary RNA alignment and consensus structure. Our method, RNAlien, starting from a single input sequence collects potential family member sequences by multiple iterations of homology search. RNA family models are fully automatically constructed for the found sequences. We have tested our method on a subset of the Rfam RNA family database. RNAlien models are a starting point to construct models of comparable sensitivity and specificity to manually curated ones from the Rfam database. RNAlien Tool and web server are available at <http://rna.tbi.univie.ac.at/rnalien/>.

INTRODUCTION

One of the basic aims of genome informatics is to annotate every single nucleotide of a genome for presence and type of biological function. The most well-known regions are protein-coding genes. The nature of non-coding RNAs (ncRNAs) and their genes has more recently started to play a role (1), with many new functions of these non-protein coding regions being elucidated using biological (2) and compu-

tational methodology (3). Of particular interest are ncRNAs which form well-defined structures that are needed to perform their function.

The sequence and the structural conservation of RNAs allows for clustering these ncRNAs into families of homologs. Structural RNA families are therefore conveniently characterized by a multiple alignment, as well as a consensus secondary structure. This allows one to trace patterns of structural conservation with covariance-preserving sequence mutations through the evolution of individual ncRNAs. For sequences that are not too far diverged, it has become a standard procedure to determine RNA family membership via computational means.

When newly sequenced genomes are to be annotated for putative functions, several tools exist that try to match a known structural RNA family to an area of the genome. The Infernal (4,5) suite of tools provides the standard machinery to match known structural RNA family models to genomic regions. The required family models are collected in the Rfam (6,7) database of more than 2000 families.

Novel RNA sequences, which are continuously discovered via next-generation sequencing experiments, are often the first known example of their RNA family. It is therefore of interest to search for homologous sequences in related species and ultimately construct a covariance model.

RNA homology search is a difficult problem (8), since simple sequence-based search can only detect very close homologs while structural conservation is needed to reliably detect remote homologs. The traditional approach would therefore combine sequence-based BLAST searches with manual inspection of each candidate in order to discard spurious hits without structural similarity.

Successful homology search for some families (9–13) that are highly variable in length and structure even requires context information like associated promoter regions. For some families even specialized homology search tools exist that consider their individual properties (14–17). Once a set

*To whom correspondence should be addressed. Tel: +49 761 203 8246; Fax: +49 761 203 7462; Email: egg@informatik.uni-freiburg.de

of diverse family members has been collected, a covariance model can be constructed from the final alignment and consensus structure. From that point on it would be possible to use the Rfam pipeline for iteratively expanding the seed alignment (14). The model can then be submitted for review, in essence repeating the steps already taken for known RNA families in the Rfam database.

The above approach is, especially up to the seed alignment, quite time-consuming and individual steps like choosing the exact start and end of the potential candidate are not standardized. In short, the model construction process would greatly profit from automation and standardization.

We now describe in detail the approach we have taken for automating the construction of a set of potentially homologous sequences given a single starting sequence, including the prediction of a common consensus secondary structure.

Our approach closely mimics a strategy that could be employed when searching for homologous sequences manually. Given that our method scales to many sequences and can be off-loaded to a web service, it aims to decrease the burden of establishing initial family models for novel sequences without much local overhead for the user.

MATERIALS AND METHODS

RNAlien is based on an iterative sequence search process. In each step new sequences from a different section of the phylogenetic tree are searched for, filtered and possibly included in the growing RNA family model. By step-wise inclusion of remote family members it is possible to increase the sensitivity for even more divergent members, without losing too much specificity.

In brief, RNAlien starts with a single sequence and optionally the organism of origin, identified by the NCBI taxonomy (18) identifier as input. An initial RNA family model is constructed from sequences found in the close taxonomic neighborhood of the input. In the second phase, the model is expanded iteratively by ascending in the taxonomic tree, and considering ever larger sub-trees, to collect family members from increasingly divergent species. When the root of the tree has been reached a final global search in each taxonomic kingdom is performed, to include sequences of interest that could not be identified before. Figure 1 shows an overview of the pipeline, a more detailed flowchart (Supplementary Figure S1) and default parameter set (Supplementary Section B – Implementation details) are available in the Supplementary Material.

Initial model construction

The goal of the initial model construction is to collect sequences that capture the secondary structure of the RNA family and some sequence variability that allows us to find more remote homologs.

RNAlien performs a sequence search via the NCBI nucleotide Blast REST interface and restricts the search to the taxonomic parent of the input organism. Using the REST interface has the advantage that the scanned databases are always up to date and that no bulk downloads are necessary.

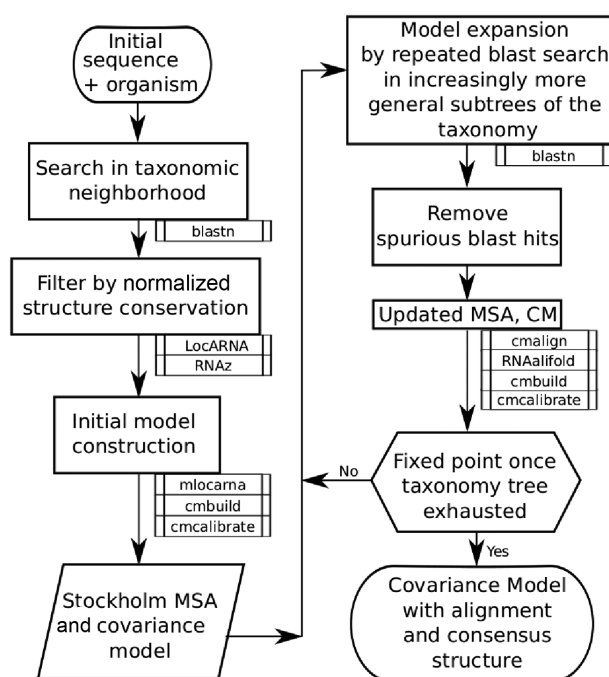


Figure 1. RNAlien program flow. RNAlien expects a single input sequence for which homology is to be established. Knowledge of the source organism provides an optional starting point in the taxonomic tree. Sequence-similar candidates are discovered (via BLAST) in closely related species with selection to reduce bias. Once a small set is discovered, an initial structural alignment and covariance model are constructed (as shown on the left) with mlocarna, cmbuild and cmcalibrate. In the second step (shown on the right side), BLAST searches continue to ever more divergent species. The covariance model is used to decide if these additional sequences are included and, if so, aligned to the model. When the whole taxonomic tree has been visited, a final search is performed and then the resulting covariance model, structural alignment and all collected sequences are returned.

BLAST hits are pre-filtered by having more than 80% coverage of the query sequence to exclude short hits. Collection of redundant hits is avoided by excluding hits with 99% or more query similarity.

Since BLAST hits are usually too short, we first expand them with flanking genomic regions (see Supplementary subsection B.5). Subsequently each candidate sequence is aligned to the input sequence using the structural RNA alignment program LocARNA (19) with a semi-global alignment in order to truncate them to the input sequence length.

The sequence identity SI is used as a measure for sequence conservation. Given the Levenshtein distance D between the input and current candidate sequence, and L , the length of the longer of the two sequences, we calculate the SI as follows: $SI = 1 - (D/L)$

Since we are interested in structural RNAs, we want to accept candidates that exhibit more structure conservation than expected for their respective sequence similarity. As a measure of structure conservation we use the SCI value introduced in the RNA gene finder RNAz (20). The SCI compares the energy $E_{\text{consensus}}$ of a consensus structure folding of the alignment A with the average energy \bar{E}_x obtained from folding each sequence x in the alignment individu-

ally, $SCI = E_{\text{consensus}}/\sqrt{E_x}$. Since the SCI depends on the sequence identity of the alignment (an alignment of identical sequences necessarily has $SCI = 1$), we normalize the SCI by the sequence identity SI of the sequences:

$$nSCI = \frac{SCI}{SI} \quad (1)$$

As a rule of thumb, alignments of structured RNA families exhibit an SCI larger than the sequence identity. We therefore accept candidates if their $nSCI > 1$.

In case the first round does not yield any acceptable candidates, we ascend in the NCBI taxonomic tree and repeat the initial model construction in the larger sub-tree.

All accepted sequences in the initial set are aligned with `mlocarna`, the multiple sequence alignment variant of `LocARNA`. The resulting structural alignment is then used to construct and calibrate a covariance model with `cm-build` and `cmcalibrate` from the `Infernal` package. We speed up calibration as described in Supplementary Material B.1 – Model construction. In the following round of the model expansion phase this model will be used to decide candidate sequence acceptance.

Model expansion

Model expansion is an iterative process depending on the family members collected so far and the corresponding covariance model.

The first step is to select representative queries for the upcoming BLAST search from the currently collected sequences. The current set is filtered, so that for all sequences with pairwise similarity greater than 95% only the first one is used. Per default the first five of these sequences are used as query sequences.

Optionally the current set can instead be clustered with the UPGMA algorithm (21), based on a distance matrix computed by `Clustal Omega` (22). `RNAlien` incrementally increases the cluster cutoff distance to form up to 5 clusters. The first sequence from each cluster is used as a query sequence. This method achieved slightly better recall in the benchmark but is optional due to the `Clustal Omega` dependency.

The target organisms are always confined to a sub-tree of the taxonomy. In each round the search space is expanded by ascending one level in the taxonomy. In order to avoid duplicates we also exclude the sub-tree of the previous round (see Supplementary Figure S2). For example, if the current taxonomic position is *Enterobacteriaceae* (family) and the previous node was *Enterobacter* (genus) all organisms that belong to *Enterobacteriaceae* but not *Enterobacter* are searched. Depending on the number of selected queries, multiple searches can be performed, the results are then pooled. The search is again performed via the REST interface of NCBI nucleotide BLAST using an E-value cutoff of 1.

To decide which of the BLAST hits to accept, we evaluate each hit with the current covariance model using `cmsearch`. To obtain E-values we set the genome size parameter of `cmsearch` to the database size of the BLAST search. At this step, we employ two different E-value cutoffs: Sequences that satisfy the strict cutoff (E-value < 0.001) are

accepted and used to build the next iteration of the covariance model. Sequences that only satisfy a relaxed cutoff of 1, are collected in a set of ‘potential’ family members and re-evaluated at the end of the pipeline using the final model.

Candidates that have been accepted are aligned to the model by `cmalign`, which creates a new Stockholm alignment. The expanded alignment may yield a slightly changed consensus structure compared to the previous iteration. We therefore recompute the consensus structure using `RNAalifold` with the recommended parameters from (23). A new model is then constructed with `cm-build` and calibrated (see Supplementary Material B.1 – Model construction) with `cmcalibrate`. Model expansion proceeds further up in the taxonomic tree until the root node has been reached.

Model finalization

In order to capture the most remote homologs, a final round analogous to model expansion, but without any taxonomic restriction is performed.

Finally, the set of potential family members collected during earlier rounds is now re-evaluated with the current model using the strict cutoff. This gives rise to the final covariance model, which is once more calibrated using `cmcalibrate`.

Model evaluation

The final covariance model and the corresponding structural alignment are inspected via `cmstat`, `RNAz`, `RNAcode` (24) and taxonomy of the included sequences. `RNAz` predicts whether the alignment contains a functional RNA structure. Since `RNAlien` is particularly geared for structural RNAs, this is an important quality indicator. `cmstat` provides additional information about the resulting covariance model itself, such as the total and effective number of sequences used to construct the model and the relative importance of sequence and structure information.

`RNAcode` predicts protein coding segments within the alignment. This allows in particular to identify RNAs that carry both functional open reading frames and RNA structure. While it is possible to use `RNAlien` for pure protein coding sequences, methods that consider protein specific features are more suited. For all found sequences a lookup at `RNAcentral` (25,26) is performed to find already existing entries. A list of `RNAcentral` identifiers is appended to the result.

The taxonomy information of the collected sequences can be useful for gaining information about the biological function of a newly isolated RNA. `RNAlien` provides a detailed log of tools and exact versions as well as intermediate results for later analysis and reproducibility of the construction process.

RESULTS AND DISCUSSION

In order to test the quality of the automatic family construction process, two different performance tests were conducted. First, we extracted a subset of RNA families from the `Rfam` 12.0 database, as detailed below. We then used `RNAlien` to reconstruct each RNA family, given a single sequence from the seed alignment. The resulting family model

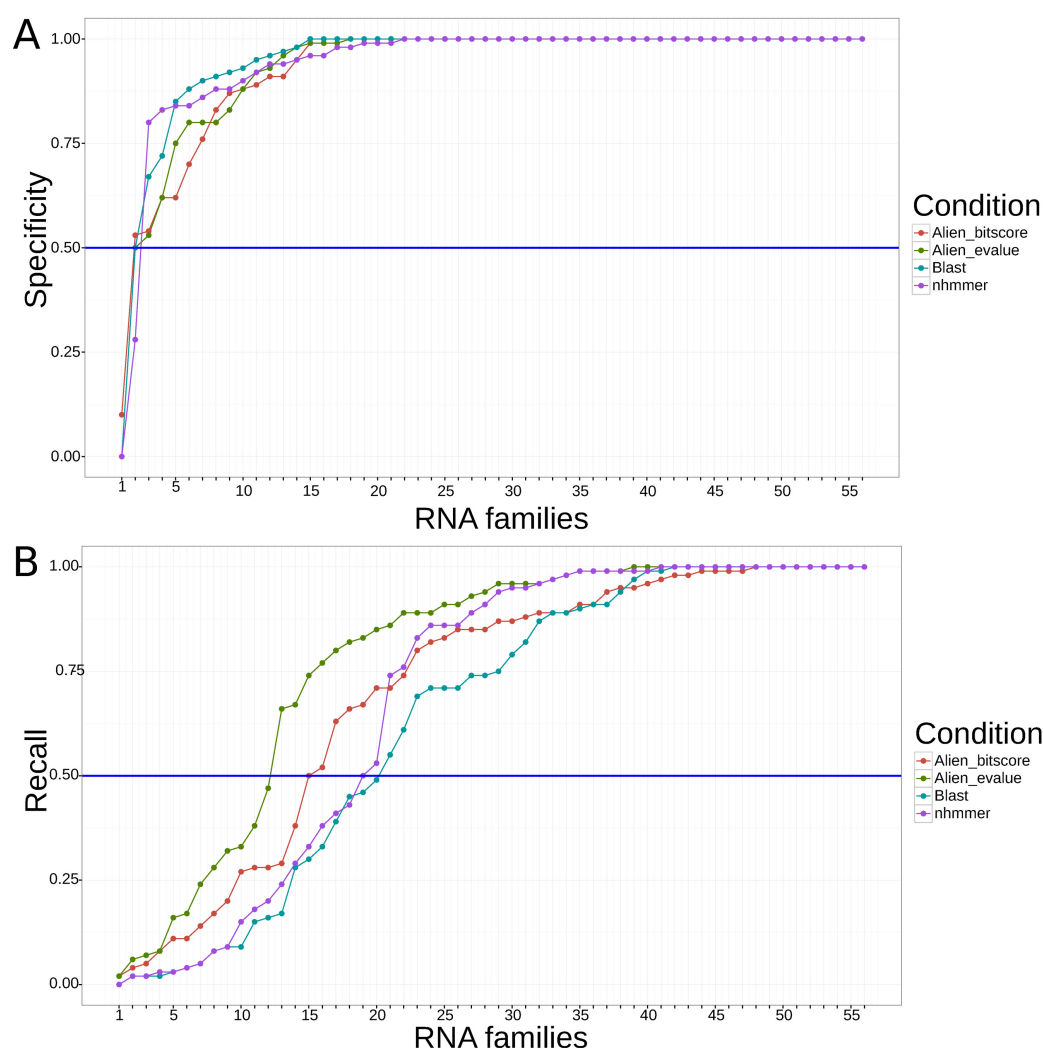


Figure 2. (A) Specificity of RNAlien homology search. The plot shows the fraction of homologs predicted by RNAlien that are recognized by the original Rfam model. In 55 of 56 cases (98%), at least half of the sequences collected by RNAlien are recognized as belonging to the Rfam model. In 35 (62%) families all sequences included by RNAlien are recognized as belonging to the Rfam model. (B) Recall of RNAlien models on Rfam sequences. We show the fraction of Rfam seed sequences recognized by the RNAlien model. In 44 of 56 cases (78%) at least half the sequences in the Rfam seed alignment are correctly recognized by the RNAlien model.

and collected sequences were compared with the original Rfam model and sequences. This test reveals the ability of RNAlien to reconstruct a known family from a single sequence.

The resulting consensus secondary structures from the first test were compared against the structure annotated in the seed alignment and the run-time for RNAlien was measured.

Second, we created a set of negative control sequences and started the model construction process. We used coding sequences, ancestral repeats, untranslated regions (UTRs) from NCBI genbank (27), Ensembl Release 83 (28), RegulonDB 9.0 (29) and random sequences. According to the procedure for structured and diverse RNA families the sequences of the negative control set were used as a input sequence for RNAlien.

Rfam families with known structure

As a test set we chose the subset of Rfam families with known structure derived from nuclear magnetic resonance or X-ray crystallography. For efficiency reasons, we discarded three families that are representing large ribosomal sub-units, each consisting of sequences exceeding 1500 nucleotides in length, leaving us with 56 families. A second test set with 192 families is contained in the Supplementary Material (see Supplementary Section D).

By arbitrary choice the first sequence of the Rfam seed-alignment was extracted and the organism of origin retrieved. This single initial sequence and the corresponding taxonomy id were used as input to RNAlien. To measure the specificity of RNAlien we tested each of the homologs predicted by RNAlien using the Rfam covariance model. RNAlien predictions that did not meet the bit score cut-

off, as described below, of the Rfam model were considered false positives.

Conversely, we measured the recall of the RNAlien model by evaluating all sequences in the Rfam seed alignment and counting all sequences not recognized by the RNAlien model as false negatives.

To provide a context for the results we performed both a BLAST and a nhmmer (30) search against the full NCBI nucleotide database with each RNAlien input sequence, without iteration. The BLAST results were aligned with mlocarna and a consensus structure was computed with RNAalifold. For the nhmmer result alignment a consensus structure was computed with RNAalifold.

The Bacterial small subunit ribosomal RNA homology search nhmmer found over 2 million hits and the resulting structural alignment was too big to further process (~600 GB) it. We therefore included it with specificity and sensitivity 1.

The resulting alignments for both tools were used to construct and calibrate a covariance model. The sequences and the model were used in the same manner as the alien result models for the benchmark.

We used two different cutoffs, one bit score based for specificity and one E-value based for the recall benchmark. The bit score cutoff uses the gathering cutoff annotated for the Rfam model to discriminate between true and false positives. However, the gathering score is quite specific for the Rfam model and is possibly not applicable to the RNAlien model.

Therefore, we used a E-value cutoff for cmsearch of 0.001 with a database size of 1000×10^6 nucleotides for families with members in eukaryotic species, corresponding to typical genome sizes. For families predominantly present in viral and prokaryotic species 1×10^6 nucleotides was set as database size.

Note, that there may well exist true homologs that are not recognized by the Rfam covariance model. Moreover, some classes of RNA, such as *RNaseP* or *SRP* RNA, are represented in Rfam by multiple families. The reported accuracies therefore present a pessimistic estimate. All intermediate results and models from this benchmark are available via <http://rna.tbi.univie.ac.at/rnalien/help#benchmark>.

A total of 55 out of 56 families (~98%) exhibit specificity > 50%, meaning that more than 50% of their sequences are recognized by the original Rfam model as family member (see Figure 2). BLAST and nhmmer achieved a slightly higher specificity than RNAlien.

In 44 of 56 cases (78%), more than 50% of the Rfam seed sequences could be categorized by the RNAlien model as a family member (see Figure 2). RNAlien has higher recall than BLAST and nhmmer.

RNA families where RNAlien performs well in terms of specificity and recall are not necessarily the same. We therefore used the minimum of recall and specificity to classify successful and poor reconstructions.

As shown in Figure 3, 43 reconstructions (~78%) achieved both recall and specificity $\geq 50\%$ and were categorized as *well reconstructed families*. In the *low recall* (recall < 50%) group 12 cases (~21%) still had specificity higher than 50%, indicating that RNAlien only found a subgroup of the Rfam family. The *low specificity* (specificity < 50%) group, consisting only of the *FMN* family, had recall above 50%. This indicates that RNAlien sometimes reports false positives.

The *Low specificity (FMN)* and *Low recall – families* groups (*Intron_gpI*, *Intron_gpII*, *Histone3*, *mir-689*, *crcB*, *c-di-GMP-II*, *THF*, *tRNA-Sec*, *Protozoa.SRP*, *group-II-D1D4-I*) are of special interest to understand problems in the model construction process. The construction processes with sub-optimal results will be discussed in the following.

The *FMN* family models the flavin mononucleotide riboswitch and the reconstructed model recovers nearly all seed sequences of the Rfam model. However, during the construction process more and more divergent hits are collected until the model becomes unspecific. In this case low

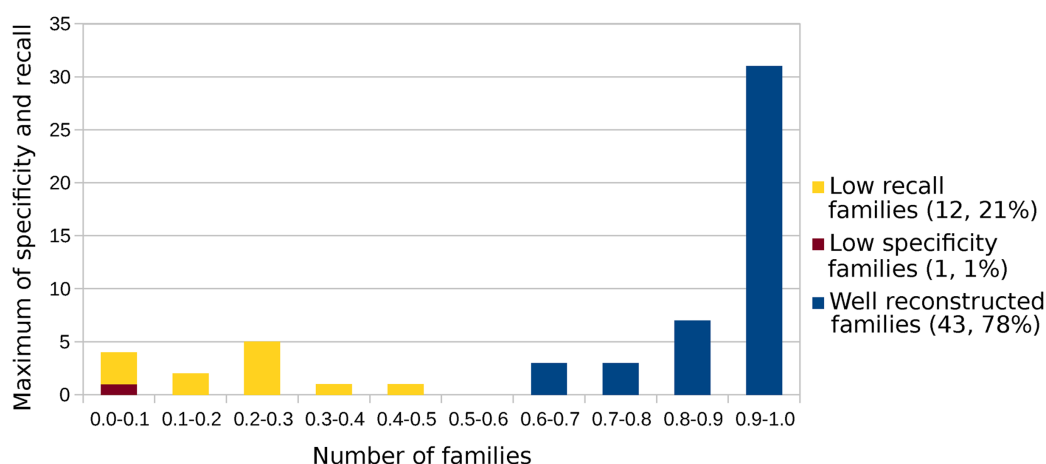


Figure 3. Family groups. To test our method, 56 Rfam family models with known structure were reconstructed by RNAlien from the first sequence picked from the family seed sequences. This plot shows the minimum of specificity and recall of all 56 reconstructed families. A total of 43 (~78%) families achieve a specificity and recall ≥ 0.5 and are referred to as group *Well reconstructed families*. Only the *FMN* family where the Rfam model detected less than 50% of the sequences collected by RNAlien (Specificity) is in the *Low specificity – families* group. A total of 12 reconstructed families (~21%) where the Alien model detected less than 50% of Rfam model seed sequences (Recall) are grouped in *Low recall – families*.

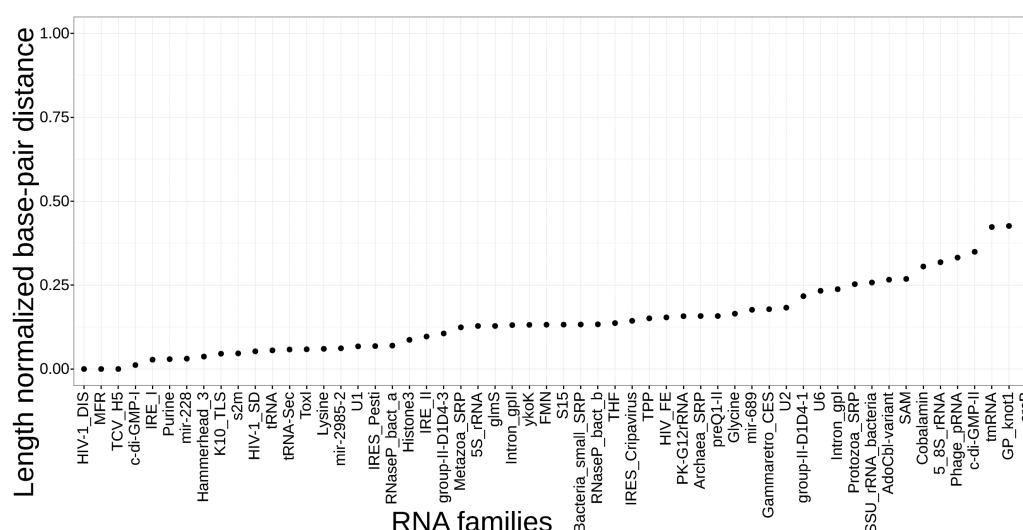


Figure 4. Length normalized secondary structure base-pair distances between the RNAlie consensus structure versus Rfam model consensus structure.

specificity is the result of an uninformative start sequence that is too short and exhibits only simple structure.

RNAlie does only recover about 47% of the Rfam seed alignment sequences for the *tRNA* family, with but these with high specificity. The family is too diverse for RNAlie to find all potential members.

The same applies to the *Protozoa_SRP* family which features related families for metazoa, as well as protozoa and to a set of other families (*Histone3*, *mir-689*, *crcB*, *c-di-GMP-II*, *THF*, *tRNA-Sec*, *group-II-D1D4-1*, *IRE_II*).

The *Intron_gplI* and *Intron_gplII* represents self splicing ribozymes that can be found in eukarya, bacteria and viruses. The *Intron_gplI* RNA features nine paired regions which are grouped in two domains, of which only the second one is featured in the Rfam model. The family is characterized by frequent variable length insertions in the loop regions.

The Rfam curators overcame this problem by manually adding biologically reasonable gaps in the seed alignment, thus reducing the cost of insertions. Moreover, the initial sequence selected for RNAlie is a viral sequence that is isolated both in terms of taxonomy and similarity with regard to the bulk of the family.

As expected, we observe among the low recall families, complex RNAs, such as group I introns, that exhibit large variation in length and would present challenges even for human experts.

Secondary structure comparison

We compared consensus secondary structures between the annotated structure for Rfam families with known 3D-structure and corresponding RNAlie alignment consensus structure. A base-pair distance, as computed by RNAdistance (31) was used for the comparison.

RNA structure distances are most meaningful when structures for sequences of equal length are compared. Both the seed alignment and the final RNAlie alignment share at least the sequence used as input for RNAlie. We processed both consensus structures before the comparison by remov-

ing all positions that map to gaps for the shared sequence. Basepairs that lose their binding position in this manner are set to be unpaired.

The resulting distances were normalized by the length of the sequence to make them comparable with each other, as shown in Figure 4. Constructions that achieved good specificity and recall in the benchmark do not necessarily have a low distance.

Running times

The running times for constructing the 56 families in above benchmarks are shown in Supplementary Figure 7. The average running time (wall-clock time) with 20 cpu-cores was about 4 h, while the fastest construction with 40 min was the *archaea_SRP* family model and the longest construction with 1 day 4 h was *Purine*.

Negative control set

In the second test we applied RNAlie on a negative data set of 651 sequences consisting of 300 random, 34 ancestral repeat, 124 coding, 193 3' and 5'-untranslated region sequences.

Homo sapiens, *Escherichia coli* and *Sulfolobus solfataricus* were used as organism of origin for 100 of the random sequences each. For none of these sequences was a second sequence search hit detected.

A total of 34 Dfam (32) families tagged as ancestral repeat a sequence was picked as input for RNAlie. The homology search for the sequences found multiple sequences but only one of the final RNAlie alignments was predicted by RNaz to be of structured RNA quality.

A total of 49 Coding sequences for *Homo sapiens*, 40 for *Escherichia coli* and 35 for *Sulfolobus solfataricus* were retrieved from Ensembl (28), RegulonDB (29) and NCBI genbank (27).

Each of the 124 sequences was used as input for RNAlie. In 24 of the cases homology search found no

Results:

Log	Fasta	Stockholm Alignment	Covariance Model	Rnaz Output	cmstat Output	Zip Archive
---------------------	-----------------------	-------------------------------------	----------------------------------	-----------------------------	-------------------------------	-----------------------------

Evaluation Results

CMstat statistics for result.cm		RNAz statistics for result alignment:	
Sequence Number	1437	Mean pairwise identity	67.83
Effective Sequences	8.12	Shannon entropy	0.57744
Consensus length	68	GC content	0.63926
Expected maximum hit-length	635	Mean single sequence minimum free energy	-31.29
Basepairs	20	Consensus minimum free energy	-29.43
Bifurcations	2	Energy contribution	-25.12
Modeltype	"cm"	Covariance contribution	-4.31
Relative Entropy CM	0.826	Combinations pair	1.83
Relative Entropy HMM	0.538	Mean z-score	-2.01
		Structure conservation index	0.94
		Background model	dinucleotide
		Decision model	structural RNA alignment quality
		SVM decision value	3.28
		SVM class propability	0.999834
		Prediction	RNA

Taxonomy overview

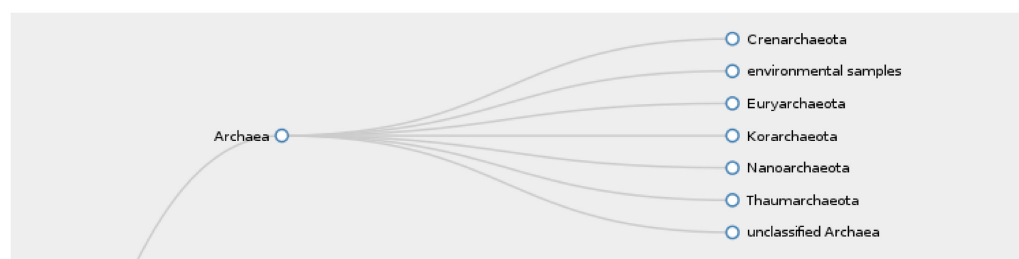


Figure 5. Result output of the RNAlie web service. The table at the top shows links to the final construction log, result sequences, alignment, covariance model, RNAz, cmstat and zip archive files. The zip archive contains all files of the construction for later reproducibility. The table in the center shows features computed for the result by cmstat, RNAz and RNACode including the prediction if the result alignment is of structural RNA alignment quality. At the bottom a slice of the taxonomic tree, including all organisms that contained hits in the construction is shown. The tree is collapsible and zoom-able for better overview.

additional hits. The 100 remaining result alignments were evaluated using RNACode, 75 of them were classified as protein coding with a P-value below 0.05. Of the remaining cases, 19 are neither predicted by RNAz to be RNA nor to be proteins by RNACode, while 6 cases were identified to be structural RNA alignments.

This means that RNAlie can, in principle, provide meaningful output when given protein coding sequences as input, with the caveat that these sequences are often too long for folding algorithms to terminate in reasonable time and that the protein-specific features (e.g. reading frame) are not used.

If RNAlie received protein coding input, this is usually indicated by RNACode in the evaluation step. Some of the constructed alignments were qualified as structured RNA by RNAz, which could be explained by conserved secondary structures that are contained in the reading frames of these alignments.

95 sequences from 5' and 3' untranslated regions from *Homo sapiens* and 98 from *Escherichia coli* were checked. *Escherichia coli* sequences are from egulonDB version 9.0 (29) *Homo sapiens* sequences are from Ensembl (28) (Release 84, GRCh38.p5), chromosome 2.

In 34 of the 193 cases no additional hits, meaning no hits that satisfied the filter criteria, were found by homology

search. 30 of these cases were UTR input sequences from *Homo sapiens*. In 114 cases, the final RNAlieN alignments were classified by RNAz as not structural RNA alignments, in 45 cases were classified as structured RNAs, of which 37 are in the 3'-UTR of *E. coli*.

Finding structured RNA in UTRs is quite expected (33). One example are terminator hairpins in prokaryotic 3'-UTRs. Possibly RNAlieN could be also used to search for structural motifs in untranslated regions.

The full table of the negative data set results can be found in Supplementary Section F - Negative control set.

As can be observed from the results in Figures 2 and 3, our models do not recover all Rfam seed sequence sets with 100% sensitivity. This is, however, completely in line with our expectations. Putative homologs are collected solely via quite stringent BLAST hits, which limits the depth of a model to those homologs to be recovered using sequence-based searches only. Additional remote homologs can be discovered by running Infernal (4).

WEB SERVER

RNA homology searches can be performed conveniently via the RNAlieN web server. The server takes a fasta sequence and the organism of origin's name or NCBI taxonomy id as input. For each iteration step the server provides information on how many sequences have been collected so far and to which node of the taxonomic tree the search has progressed.

Upon completion the sequences, structural alignment and calibrated covariance model are available via download links (see Figure 5). All intermediate results are available as compressed archives for documentation and review of the results.

A key feature of the web server is a zoom-able and collapse-able taxonomic tree of the organisms where family members were found. The results of model evaluation, like the cmstat, RNAz and RNAcode output are summarized in a table.

The final covariance model can be directly passed on to the CMCompare web service (34,35) which compares it to all RNA family models in the Rfam database. This allows to find related families, or even an alternative pre-existing family model for the newly constructed model.

CONCLUSION

With RNAlieN we provide an automated pipeline for RNA homology search. Starting from a single sequence, a combined sequence-structure alignment is constructed. Sequences are collected from an ever-wider search within the phylogeny of the starting sequence, with the goal of producing a family of phylogenetically diverse members. The resulting family comes complete with a set of statistical predictors of quality, and a covariance model for further searches.

These results show that our method does indeed produce models that may serve as initial seed models for further investigation. The resulting alignment could also be used as input for iteratively expanding input seed alignments via multiple rounds (14).

However there are RNA families (10–13) for which an automated approach can only partially succeed, because the RNAs exhibit large variation in length and structure. Here, the use of contextual information, like promoter binding sites and other expert knowledge can help.

We point out that the dependency on BLAST could be easily dropped by directly using cmsearch for candidate search. While this could improve sensitivity, it would incur much higher computational cost, especially when scanning eukaryotic genomes. In the future we plan to add candidate search via nhmmer, speed up the pipeline by modifying model calibration and expand the construction process to include alternative model concepts (36).

Together with the web server, RNAlieN provides a completely automated and easy to use method to construct initial structured RNA family models, based on a single initial sequence. This in turn considerably reduces the workload of an investigation into a novel sequence whose pedigree is unknown.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

Austrian Fonds zur Förderung der wissenschaftlichen Forschung (FWF, in part); project 'Doktoratskolleg RNA Biology [W1207-B09]'; project 'SFB F43 RNA regulation of the transcriptome'; Swiss National Science Foundation (SNSF) [project CRSII3_154471/1 to I.H.]; Deutsche Forschungsgemeinschaft (DFG) [BA 2168/4-3 SPP 1395 InKoMBio]. Funding for open access charge: Austrian Fonds zur Förderung der wissenschaftlichen Forschung (FWF, in part); project 'Doktoratskolleg RNA Biology [W1207-B09]'; project 'SFB F43 RNA regulation of the transcriptome'. The authors also thank the anonymous referees for their helpful comments.

Conflict of interest statement. None declared.

REFERENCES

- Mattick, J.S. and Makunin, I.V. (2006) Non-coding RNA. *Hum. Mol. Genet.*, **15**, R17–R29.
- Hüttenhofer, A. and Vogel, J. (2006) Experimental approaches to identify non-coding RNAs. *Nucleic Acids Res.*, **34**, 635–646.
- Washietl, S., Will, S., Hendrix, D.A., Goff, L.A., Rinn, J.L., Berger, B. and Kellis, M. (2012) Computational analysis of noncoding RNAs. *Wiley Interdiscip. Rev.*, **3**, 759–778.
- Nawrocki, E.P., Kolbe, D.L. and Eddy, S.R. (2009) Infernal 1.0: inference of RNA alignments. *Bioinformatics*, **25**, 1335–1337.
- Nawrocki, E.P. and Eddy, S.R. (2013) Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, **29**, 2933–2935.
- Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A. and Eddy, S.R. (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.
- Gardner, P.P., Daub, J., Tate, J., Moore, B.L., Osuch, I.H., Griffiths-Jones, S., Finn, R.D., Nawrocki, E.P., Kolbe, D.L., Eddy, S.R. et al. (2011) Rfam: Wikipedia, clans and the 'decimal' release. *Nucleic Acids Res.*, **39**(Suppl. 1), D141–D145.
- Menzel, P., Gorodkin, J. and Stadler, P.F. (2009) The tedious task of finding homologous noncoding RNA genes. *RNA*, **15**, 2075–2082.
- Li, S.-C., Pan, C.-Y. and Lin, W.-C. (2006) Bioinformatic discovery of microRNA precursors from human ESTs and introns. *BMC Genomics*, **7**, 1–11.

10. Stadler, P.F., Chen, J. J.-L., Hackermüller, J., Hoffmann, S., Horn, F., Khaitovich, P., Kretschmar, A.K., Mosig, A., Prohaska, S.J., Qi, X. *et al.* (2009) Evolution of vault RNAs. *Mol. Biol. Evol.*, **26**, 1975–1991.
11. Gruber, A.R., Koper-Emde, D., Marz, M., Tafer, H., Bernhart, S., Obernosterer, G., Mosig, A., Hofacker, I.L., Stadler, P.F. and Benecke, B.-J. (2008) Invertebrate 7SK /textit{snRNAs}. *J. Mol. Evol.*, **66**, 107–115.
12. Gruber, A.R., Kilgus, C., Mosig, A., Hofacker, I.L., Hennig, W. and Stadler, P.F. (2008) Arthropod 7SK RNA. *Mol. Biol. Evol.*, **25**, 1923–1930.
13. Boria, I., Gruber, A.R., Tanzer, A., Bernhart, S.H., Lorenz, R., Mueller, M.M., Hofacker, I.L. and Stadler, P.F. (2010) Nematode *sbRNAs*: homologs of vertebrate YRNAs. *J. Mol. Evol.*, **70**, 346–358.
14. Nawrocki, E.P., Burge, S.W., Bateman, A., Daub, J., Eberhardt, R.Y., Eddy, S.R., Floden, E.W., Gardner, P.P., Jones, T.A., Tate, J. *et al.* (2015) Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.*, **43**, D130–D137.
15. Lowe, T.M. and Eddy, S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.
16. Lagesen, K., Hallin, P., Rødland, E., Stærfeldt, H., Rognes, T. and Ussery, D. (2007) RNAmmer: consistent annotation of rRNA genes in genomic sequences. *Nucleic Acids Res.*, **35**, 3100–3108.
17. Lowe, T.M. and Eddy, S.R. (1999) A computational screen for methylation guide *snoRNAs* in yeast. *Science*, **283**, 1168–1171.
18. Wheeler, D.L., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Edgar, R., Federhen, S. *et al.* (2007) Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, **35**(Suppl. 1), D5–D12.
19. Will, S., Reiche, K., Hofacker, I.L., Stadler, P.F. and Backofen, R. (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, e65.
20. Gruber, A.R., Findeiß, S., Washietl, S., Hofacker, I.L. and Stadler, P.F. (2010) RNAz 2.0: Improved noncoding RNA detection. In: *Biocomputing 2010: Proceedings of the Pacific Symposium, Kamuela, Hawaii, USA, 4–8 January 2010*. pp. 69–79.
21. Sokal, R. and Michener, C. (1958) A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.*, **38**, 1409–1438.
22. Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J. *et al.* (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, **7**, 1–6.
23. Bernhart, S.H., Hofacker, I.L., Will, S., Gruber, A.R. and Stadler, P.F. (2008) RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, **9**, 474–487.
24. Washietl, S., Findeiß, S., Müller, S.A., Kalkhof, S., von Bergen, M., Hofacker, I.L., Stadler, P.F. and Goldman, N. (2011) RNACode: robust discrimination of coding and noncoding regions in comparative sequence data. *RNA*, **17**, 578–594.
25. Bateman, A., Agrawal, S., Birney, E., Bruford, E.A., Bujnicki, J.M., Cochran, G., Cole, J.R., Dinger, M.E., Enright, A.J., Gardner, P.P. *et al.* (2011) RNAcentral: a vision for an international database of RNA sequences. *RNA*, **17**, 1941–1946.
26. Consortium, RNA (2014) RNAcentral: an international database of ncRNA sequences. *Nucleic Acids Res.*, **43**, D123–D129.
27. Benson, D.A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. and Sayers, E.W. (2011) GenBank. *Nucleic Acids Res.*, **39**, D32–D37.
28. Yates, A., Akanni, W., Amode, M.R., Barrell, D., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Fitzgerald, S., Gil, L. *et al.* (2016) Ensembl 2016. *Nucleic Acids Res.*, **44**, D710–D716.
29. Salgado, H., Peralta-Gil, M., Gama-Castro, S., Santos-Zavaleta, A., Muñoz-Rascado, L., García-Sotelo, J.S., Weiss, V., Solano-Lira, H., Martínez-Flores, I., Medina-Rivera, A. *et al.* (2013) RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Res.*, **41**, D203–D213.
30. Wheeler, T.J. and Eddy, S.R. (2013) nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, **29**, 2487–2489.
31. Lorenz, R., Bernhart, S. H.F., zu Siederdissen, C.H., Tafer, H., Flamm, C., Stadler, P.F. and Hofacker, I.L. (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**, 26–40.
32. Wheeler, T.J., Clements, J., Eddy, S.R., Hubley, R., Jones, T.A., Jurka, J., Smit, A.F. and Finn, R.D. (2013) Dfam: a database of repetitive DNA based on profile hidden Markov models. *Nucleic Acids Res.*, **41**, D70–D82.
33. Pichon, X., Wilson, L.A., Stoneley, M., Bastide, A., King, H.A., Somers, J. and Willis, A.E. (2012) RNA binding protein/RNA element interactions and the control of translation. *Curr. Protein Pept. Sci.*, **13**, 294–304.
34. Eggenhofer, F., Hofacker, I.L. and zu Siederdissen, C.H. (2013) CMCompare webserver: comparing RNA families via covariance models. *Nucleic Acids Res.*, **41**, 499–503.
35. Höner zu Siederdissen, C. and Hofacker, I.L. (2010) Discriminatory power of RNA family models. *Bioinformatics*, **26**, i453–i459.
36. Janssen, S. and Giegerich, R. (2015) Ambivalent covariance models. *BMC Bioinformatics*, **16**, 178–195.

RNAlien - Unsupervised RNA family model construction - Supplement

Florian Eggenhofer^{1,2}, Ivo L. Hofacker^{1,3} and Christian Höner zu
Siederdisen^{4,1,5},

¹ Institute for Theoretical Chemistry, University of Vienna, Währingerstrasse
17,A-1090 Vienna, Austria

² Bioinformatics Group, Department of Computer Science University of Freiburg,
Georges-Köhler-Allee ,79110 Freiburg, Germany

³ Bioinformatics and Computational Biology research group, University of Vienna,
Währingerstrasse 17,A-1090 Vienna, Austria

⁴ Bioinformatics Group, Department of Computer Science, University of Leipzig,
D-04107 Leipzig

⁵ Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße
16-18, D-04107 Leipzig, Germany

Table of Contents

RNAlien - Unsupervised RNA family model construction - Supplement . .	i
<i>Florian Eggenhofer, Ivo L. Hofacker and Christian Höner zu Siederdisen</i>	
A RNAlien detailed flowchart	iii
B Implementation Details	iv
B.1 Initial model construction	iv
Search:	iv
Filtering hits:	iv
Model Construction:	vi
Select Queries:	vi
B.2 Model expansion	vii
Search:	vii
Filtering hits:	vii
Model Construction:	viii
Select Queries:	ix
B.3 Model finalization:	ix
Search, Filter:	x
Reevaluation of potential candidates:	x
Modelconstruction:	x
B.4 Model evaluation	x
B.5 Blast hit extension	xi
C Rfam RNA families with known structure	xii
D Diverse Rfam RNA families benchmark set	xvi
E Negative control set	xxiii
E.1 Random sequences	xxiii
E.2 Ancestral repeats	xxiii
E.3 Coding sequences	xxiii
E.4 UTR regions	xxiii

A RNAlien detailed flowchart

Detailed flowchart representation of the RNAlien program flow.

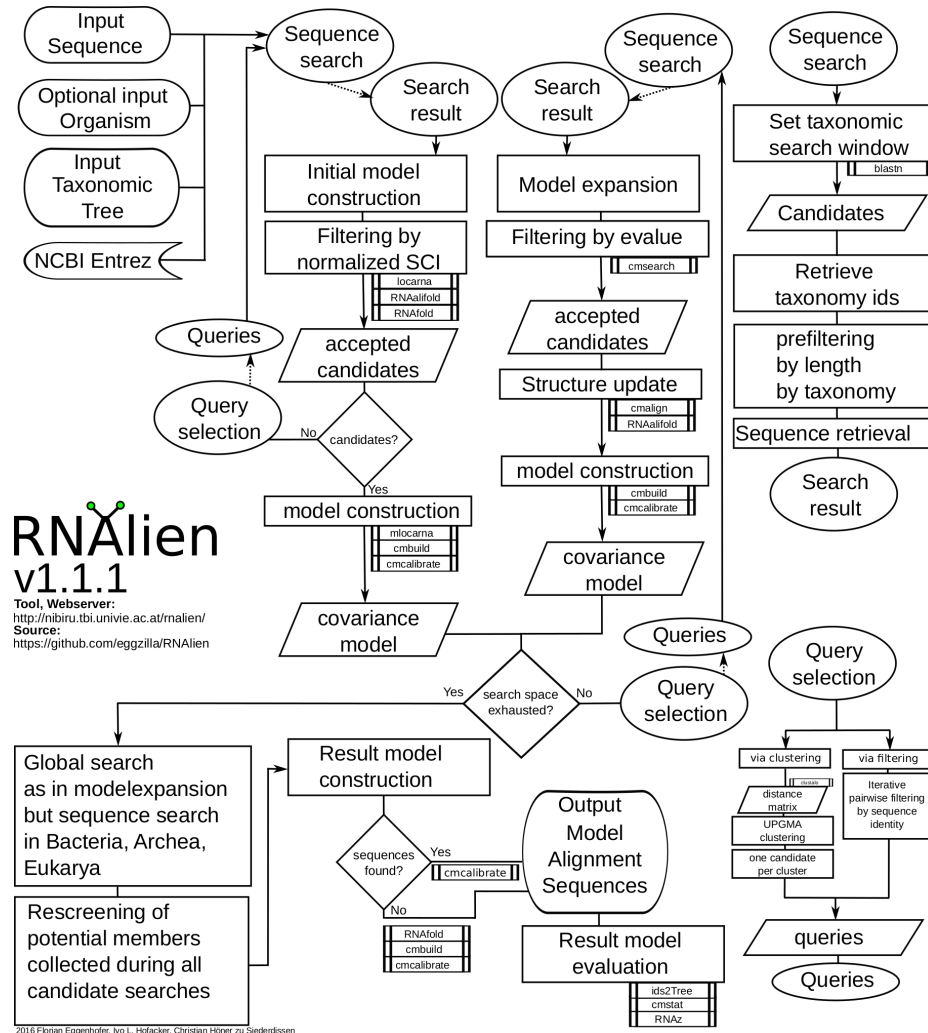


Fig. 1. Detailed program flow chart for RNAlien program.

B Implementation Details

RNAlien depends on several external tools and interfaces, which are listed in this section. System and function calls are included with their parameters and highlighted in *italic*. A starting point in the taxonomic tree is set, either specified by the input NCBI taxonomy id, or by running a nucleotide **BLAST** search via the NCBI REST interface and selecting the organism of the best hit. The model construction process starts at this organism and performs a initial model construction step. **RNAlien** retrieves the taxonomic lineage of starting organism from the NCBI ENTREZ REST interface. After each of these steps **RNAlien** proceeds to the taxonomic parent of the current taxonomic node. If a model was already constructed in a previous step then a model expansion step is performed, otherwise a initial model construction is reattempted. Once the root of the taxonomic tree has been reached model expansion stops and the model finalization step is performed.

B.1 Initial model construction

RNAlien tries to establish an initial set of sequences related to the input sequence, that serve as seed for further expansion of the model.

Search: Candidate search is performed via the nucleotide **BLAST** REST interface which returns a list of hits. The organisms to be searched are restricted by the current taxonomic node of the step in two ways. To avoid overenrichment of sequences similar to included ones, already visited organisms are excluded. Only organisms that are associated with children of the current node are searched. For example if the current taxonomic position is *Enterobacteriaceae* and the previous node was *Enterobacter* all other organisms that belong to *Enterobacteriaceae* excluding *Enterobacter* are searched.

Summary of NCBI BLAST REST function call (one for each query, all other parameters default):

blastHTTP	
<i>parameter</i>	<i>value</i>
program	blastn
database	nt
querySequence	currentsequence
hitlistSize	5000
e-value	0.001
uppertaxonomylimit	currenttaxonomyid
lowertaxonomylimit	previoustaxonomyid

Filtering hits: BLAST hits are filtered by consecutively by following criteria: Hit has to achieve over 80% coverage of the query sequence.

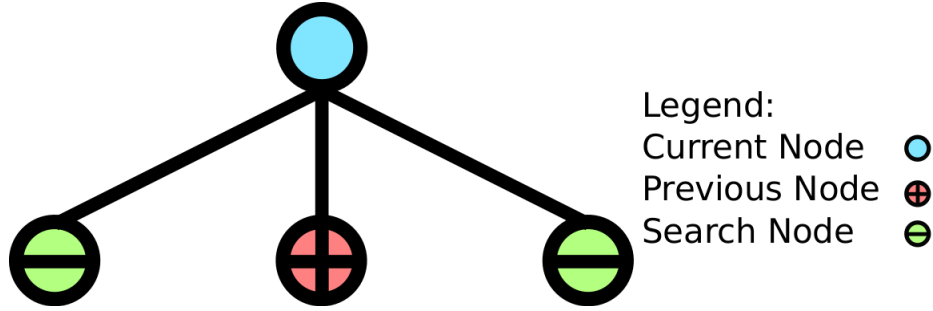


Fig. 2. Organisms used for candidate search are determined as follows. All organisms and their corresponding genomes that are associated with the currently selected position in the taxonomic tree are used for searching. Excepted from this are organisms that have already been searched in previous rounds.

The hit must not exceed query length by factor of three.

Similarity of the hit to the query must be under 99%.

The remaining hits are expanded to query length as explained in subsection B.5. The gene id contained in the **BLAST** result and the expanded coordinates are used to retrieve nucleotide sequence from the Entrez REST interface. The sequences are filtered by normalized structure conservation index (nSCI). To compute the nSCI for each candidate sequence we need the minimum free secondary structure folding energy (MFE) of the candidate and the input sequence which is computed with RNAfold.

RNAfold	
<i>parameter</i>	<i>value</i>
--noPS	
inputfilePath	fastaFilePath
outputFilepath	foldFilePath

Furthermore the structure conservation index and the sequence identity of the candidate and input sequence are required. The candidate sequence is pair-wise aligned with free end-gap setting (semi-globally) to the input sequence. For each of these alignments the structure conservation index SCI is computed via RNAalifold.

locarna	
<i>parameter</i>	<i>value</i>
--write-structure	
--free-endgaps=+ + --	
--clustal	clustalFormatFilePath
inputFilepath1	inputFastaFilePath
inputFilepath2	inputFastaFilePath
outputFilepath	locarnaFilePath

RNAalifold	
<i>parameter</i>	<i>value</i>
inputFilePath	clustalFormatFilePath
outputFilePath	aliFoldFilePath

The sequence identity is computed via levenstein distance with following edit costs (delete,insert,substitution,transposition)=1. Candidate sequences are accepted for model construction if their nSCI exceeds one.

Model Construction: Candidate sequences that passed the nSCI filter are then used to build the initial model together with the input sequence. The sequences are structually aligned with mlocarna.

mlocarna	
<i>parameter</i>	<i>value</i>
inputFilePath	inputFastaFilePath
outputFilePath	mlocarnaFilePath

cmbuild is applied to the resulting structural stockholm alignment to construct a covariance model.

cmbuild	
<i>parameter</i>	<i>value</i>
--refine	
inputModelFilePath	cmFilePath
inputAlignmentFilePath	stockholmAlignmentFilePath
outputLogFilePath	logFilePath

The covariance model is used in the model expansion rounds to filter candidates and is therefore calibrated with cmcalibrate. This step is very time-consuming but sped up by using nonstandard (**--beta** 10^{-4}) parameter. This affects the pre-filter steps of **cmsearch**, but not the final step where the sequence is aligned to the model via the CYK algorithm. Meaning that this increase in calibration speed reduces sensitivity but not specificity.

cmcalibrate	
<i>parameter</i>	<i>value</i>
--beta 1E-4	
inputModelFilePath	cmFilePath
outputFilePath	mlocarnaFilePath

Select Queries: At the end of the round queries for the candidate search of the next round are selected. **RNAlien** features a filtering and a clustering based method of query selection.

Filtering based method is the default method and iteratively removes all entries from the list of collected sequences, that do not have at most 95% pairwise sequence identity. This method has less specificity and sensitivity in the benchmarks (see 5, 6), but it is faster and removes the dependency on **clustalo**.

Clustering based method can alternatively be used by supplying the `-m` commandline switch with the value *clustering* to **RNAlien**. Clustal omega is used to compute a pairwise distance matrix of all collected sequences for clustering.

clustalo	
<i>parameter</i>	<i>value</i>
<code>--full</code>	
<code>--distmat-out</code>	<code>matrixFilePath</code>
<code>--infile</code>	<code>fastaFilePath</code>
<code>outputFilePath</code>	<code>clustaloFilePath</code>

RNAlien clusters the sequences via *unweighted pair group method with arithmetic mean* (UPGMA) and then incrementally increases the cutoff distance until 5 clusters can be formed. If less than 5 sequences have been collected, then each of them will be used as query.

B.2 Model expansion

After a initial model has been constructed **RNAlien** enters into model expansion phase.

Search: Searching is performed as described in Initial model construction but with a relaxed e-value cutoff of 1 during the **BLAST** search.

blastHTTP	
<i>parameter</i>	<i>value</i>
<code>program</code>	<code>blastn</code>
<code>database</code>	<code>nt</code>
<code>querySequence</code>	<code>currentsequence</code>
<code>hitlistSize</code>	<code>5000</code>
<code>e-value</code>	<code>1</code>
<code>uppertaxonomylimit</code>	<code>currenttaxonomyid</code>
<code>lowertaxonomylimit</code>	<code>previoustaxonomyid</code>

Filtering hits: Filtering of **BLAST** hits and hit expansion is performed as described in Initial model construction.

Sequences are also retrieved via the NCBI Entrez REST interface but then filtered with a different approach. We use the calibrated covariance model of the previous round and apply it with `cmsearch` to the candidate sequences. Candidates are accepted into the growing model if their e-value is below 0.001 or as specified by the `inputValueCutoff` commandline argument.

To ensure a meaningful e-value cutoff we need to consider the size of the database. We reuse the size of the blast database the hit originates from.

The value is not by itself contained in the blast XML output, but all the parameters needed to compute it. The relationship of E-value and bitscore (Equation 3 adopted from [1]):

$$e = d * q * 2^{-b} \quad (1)$$

where d = databasesize

e = e-value

b = bitscore

q = querylength

We compute the database size in Mbases that was used for the blast search as follows, by rearranging the equation above:

$$d = (e * 2^b) / q \quad (2)$$

where d = databasesize

e = e-value

b = bitscore

q = querylength

Candidates are accepted into the growing model if their `cmsearch` E-value is below 0.001 or as specified by the `inputValueCutoff` commandline argument

cmsearch	
<i>parameter</i>	<i>value</i>
--notrunc	
-Z	databaseSize
-g	covarianceModelPath
inputFilePath	sequenceFilePath
outputFilePath	cmsearchFilePath

Model Construction: Candidates that were accepted by `cmsearch` and already collected sequences are structurally aligned with the covariance model of the previous round.

cmalign	
<i>parameter</i>	<i>value</i>
inputModelFilePath	cmFilePath
inputSequenceFilePath	fastaFilePath
outputAlignmentFilePath	stockholmAlignmentFilePath

As the secondary structure of the resulting stockholm alignment is not updated in this process, a consensus secondary structure of the new alignment is computed via RNAalifold, with settings specifically optimized to consider covariance contributions. The old consensus secondary structure is replaced with the new one in the alignment.

RNAalifold	
<i>parameter</i>	<i>value</i>
-r	
--cfactor	
-Z	databaseSize
-g	covarianceModelPath
inputFilepath	sequenceFilePath
outputFilepath	cmsearchFilePath

cmbuild is used to construct a updated covariance model.

cmbuild	
<i>parameter</i>	<i>value</i>
--refine	
inputModelFilepath	cmFilePath
inputAlignmentFilepath	stockholmAlignmentFilePath
outputLogFilepath	logFilePath

The model is calibrated with cmcalibrate for the following candidate search.

cmcalibrate	
<i>parameter</i>	<i>value</i>
--beta 1E-4	
inputModelFilepath	cmFilePath
outputFilepath	mlocarnaFilePath

Select Queries: Search candidates for the next round are selected as described in Initial model construction.

B.3 Model finalization:

Model finalization serves to collect family members that could not be included in earlier rounds, because the model was too specific at that point and make the results available for the user. First individual candidate searches are performed in Archea, Bacteria, and Eukaria or as specified by the taxonomyRestriction commandline argument. The results are pooled and then processed as described in model expansion. The resulting model is then used to reevaluate collected potential candidates. These sequences are filtered as described in model expansion and if accepted included into the model. This final model is then calibrated with

default options to make it immediately useable for further homology search by the user.

Search, Filter: as in modelexpansion for 3 kingdoms (Archea - taxid 2157, Bacteria - taxid 2, Eukaria - taxid 2759)
Modelconstruction as in Modelexpansion

Reevaluation of potential candidates: Filter like in Modelexpansion

Modelconstruction: as described above

cmbuild	
<i>parameter</i>	<i>value</i>
--refine	
inputModelFilepath	cmFilePath
inputAlignmentFilepath	stockholmAlignmentFilePath
outputLogFilepath	logFilePath

Calibration is done without speedup by --beta 1E-4 for the final model

cmcalibrate	
<i>parameter</i>	<i>value</i>
inputModelFilepath	cmFilePath
outputFilepath	mlocarnaFilePath

B.4 Model evaluation

In this step descriptors for the result files are computed. The covariance model is used as input for **cmstat**, which computes among other features the cm and hmm content of the model. **cmalign** is used to generate a **clustalw** format result alignment which is prefiltered by **rnazSelectSeqs.pl** (auxiliary script packaged with **RNAz**). This filtered alignment is used as input for **RNAz** set to use the decision model for structural alignments. The most relevant output of **RNAz** in this case is if it predicts the input to be structured RNA, which is a indicator for successful model constructions.

cmalign	
<i>parameter</i>	<i>value</i>
--outformat=Clustal	
inputModelFilepath	cmFilePath
outputFilepath	mlocarnaFilePath

rnazSelectSeqs.pl	
<i>parameter</i>	<i>value</i>
inputFilePath	clustalFilePath
outputFilePath	selectedClustalFilePath

RNAz	
<i>parameter</i>	<i>value</i>
	-1
inputFilePath	selectedClustalFilePath
outputFilePath	rnazFilePath

cmstat	
<i>parameter</i>	<i>value</i>
	-1
inputFilePath	covarianceModelPath
outputFilePath	cmstatFilePath

B.5 Blast hit extension

RNAlien expands found **BLAST** hits to the query length if possible.

Same strand **BLAST** hit are extended as follows,

$$\begin{aligned}
 t &= h - q \\
 T &= H + (L - Q) \\
 s(t) &= \begin{cases} t, & \text{if } t \geq 0 \\ 0, & \text{otherwise} \end{cases} \\
 E(T) &= \begin{cases} b, & \text{if } T \geq b \\ T, & \text{otherwise} \end{cases}
 \end{aligned}$$

where h is the start coordinate of the hit, t is the extended start coordinate, q is the start coordinate of the hit on the query, H is the end coordinate of the hit, T is the extended endcoordinate, Q is the end coordinate of the hit on the query, L is the length of the query sequence b is the length of the sequence the hit maps to s is the start coordinate of the extended sequence checked for being within the available coordinates of the hit sequence, E is the end coordinate of the extended sequence checked for being within the available coordinates of the hit sequence



Fig. 3. Extension of BLAST hit and query on the same strand to query length, where h is the start coordinate of the hit, t is the extended start coordinate, q is the start coordinate of the hit on the query, H is the end coordinate of the hit, T is the extended endcoordinate, Q is the end coordinate of the hit on the query, L is the length of the query sequence b is the length of the sequence the hit maps to s is the start coordinate of the extended sequence checked for being within the available coordinates of the hit sequence, E is the end coordinate of the extended sequence checked for being within the available coordinates of the hit sequence

Different Strand BLAST hit are extended as follows,

$$t = h + q$$

$$T = H - (L - Q)$$

$$s(t) = \begin{cases} b, & \text{if } t \geq b \\ t, & \text{otherwise} \end{cases}$$

$$e(T) = \begin{cases} T, & \text{if } T \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where h is the start coordinate of the hit,
 t is the extended start coordinate,
 q is the start coordinate of the hit on the query,
 H is the end coordinate of the hit,
 T is the extended endcoordinate,
 Q is the end coordinate of the hit on the query,
 L is the length of the query sequence
 b is the length of the sequence the hit maps to
 s is the start coordinate of the extended sequence checked for being within the available coordinates of the hit sequence,
 E is the end coordinate of the extended sequence checked for being within the available coordinates of the hit sequence

C Rfam RNA families with known structure

This section contains additional plots for the RNA families with known structure featured in the paper. The first 2 plots show the changes of specificity and

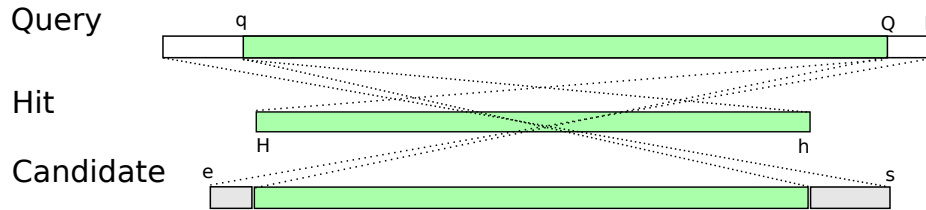


Fig. 4. Extension of BLAST hit and query on different strands to query length

sensitivity after subsequently applying the suggestions of the reviewers. The original version before the review was RNAlien 1.0.0, the one including all changes listed here has version 1.1.1.

Inclusion of paralogs and toggling of the refine switch for cmbuild were included first, this has improved both specificity, as well as recall. Additionally to this, we changed the method for selecting queries for searching candidates from clustering all collected sequences and picking one sequence per cluster to filtering all sequence that do not have a pairwise sequence identity of less than 95%.

While the specificity is slightly only lower, there is a decrease in specificity. Nevertheless we have selected the new query selection method as default, because it is substantially faster and it drops the dependency on clustal-omega.

Blast hits are now also checked for the hit to have at least 80% coverage of the query. This feature should have been included in RNAlien 1.0.0, but was faulty.

Query sequences submitted to blast can be softmasked with conservation information from /cmaalign. This feature is not considered in the shown benchmarks, but can be activated via commandline switch.

All of the newly introduced features can be controlled via commandline switches, with exception of the cmbuild refinement.

The runtime of RNAlien for the structured RNA test set

Following is the table of sequences from the Rfam 12.0 seed alignments of families with known structure, that was used in the result section. The first sequence of the family was picked with the exception of sequences that are associated with metagenomic tax ids that could not be processed by the NCBI REST BLAST interface.

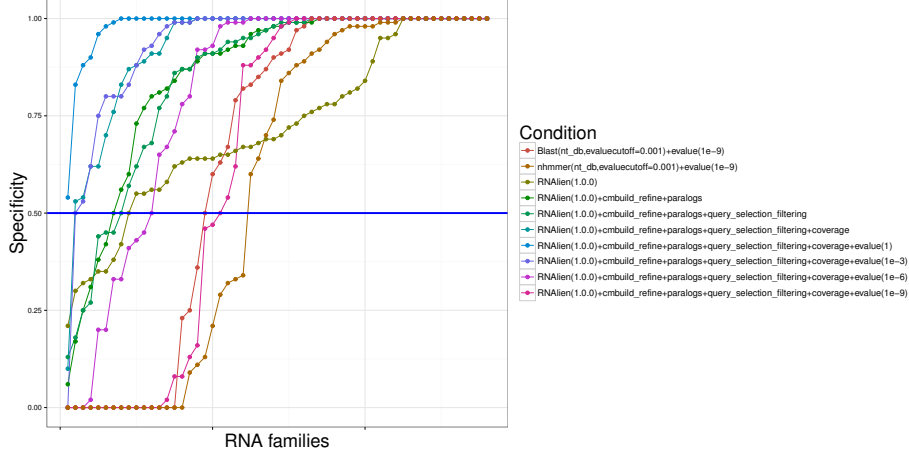


Fig. 5. Specificity for 56 RNA families with known 3D structure.

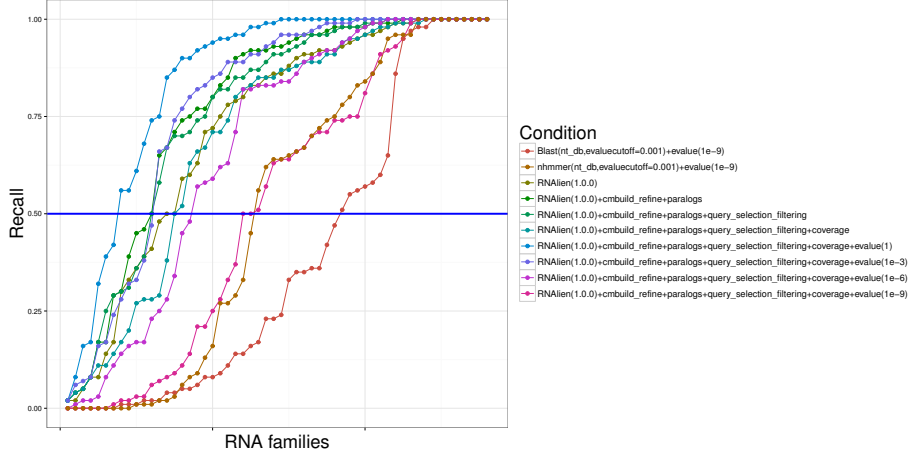


Fig. 6. Recall for 56 RNA families with known 3D structure.

Table 1: RNA families with known structure benchmark table. Column names A to N are placeholders for the following names: Specificity_Alien (=A) Sens_Alien (=B) Spec+paralogs+refine (=C) Sens+paralogs+refine (=D) Spec+filterings (=E) Sens+filtering (=F) Spec+coverage (=G) Sens+coverage (=H) Spec_value (=I) Sens_value (=J) Spec_nhmmer_value (=K) Sens_nhmmer_value (=L) Spec_blast_value (=M) Sens_blast_value (=N). The column names annotated with value were computed with a value cutoff of 10^{-3} and a databasesize of 10^9 bases per default, with the exception of families that can be found exclusively in prokaryotes and viruses.

Rfam id	Rfam name	A	B	C	D	E	F	G	H	I	J	K	L	M	N
5S_rRNA	RF00001	0.64	0.91	0.99	0.92	0.77	0.82	1	0.87	1	0.83	0.94	0.53	0.72	0.55

Continued on next page

Table 1 – continued from previous page

Rfam id	Rfam name	A	B	C	D	E	F	G	H	I	J	K	L	M	N
5.8S_rRNA	RF00002	0.95	0.9	1	0.9	1	0.85	1	0.85	1	0.89	0.96	0.95	1	0.74
U1	RF00003	0.58	0.88	0.97	1	0.99	1	1	1	1	0.99	0.86	0.99	1	0.75
U2	RF00004	0.62	0.99	0.89	0.98	0.99	0.99	0.99	0.95	0.99	0.96	0.84	0.99	0.96	0.89
tRNA	RF00005	0.77	0.48	1	0.75	1	0.7	1	0.63	0.75	0.47	0.9	0.15	1	0.04
Hammerhead_3	RF00008	1	0.63	1	0.74	1	0.74	1	0.74	1	0.74	1	0.74	1	0.74
RNaseP_bact_a	RF00010	0.56	1	0.93	0.98	0.94	0.98	1	1	1	1	0.98	1	1	1
RNaseP_bact_b	RF00011	0.55	1	0.99	1	1	1	1	1	1	1	0.99	1	0.91	1
Metazoa_SRP	RF00017	0.35	0.92	0.06	0.96	0.13	0.96	1	0.99	1	0.99	0.95	0.96	1	0.99
tmRNA	RF00023	0.65	0.92	0.98	0.93	0.98	0.92	0.99	0.88	0.99	0.91	0.98	0.95	0.98	0.61
U6	RF00026	0.64	0.83	0.98	0.83	0.99	0.82	1	0.82	1	0.8	0.88	0.89	0.93	0.71
Intron_gpI	RF00028	0.32	0.08	0.17	0.17	0.27	0.25	1	0.08	1	0.08	0.28	0.08	1	0.08
Intron_gpII	RF00029	0.75	0.41	0.31	0.65	0.92	0.58	0.89	0.2	0.92	0.16	0.84	0.2	1	0.09
Histone3	RF00032	1	0.02	1	0.02	1	0.02	1	0.02	0.5	0.02	1	0.02	1	0.02
IRE_I	RF00037	0.78	0.92	0.99	0.92	0.95	0.87	0.95	0.89	0.96	0.89	1	0.05	1	0.05
Phage_pRNA	RF00044	0.8	1	0.8	1	0.8	1	1	1	1	1	0.8	1	1	1
FMN	RF00050	0.21	0.79	0.6	1	0.18	1	0.1	0.66	1	1	1	1	0.85	1
TPP	RF00059	0.63	0.83	0.77	0.92	0.57	0.91	0.87	0.83	0.8	0.82	1	0.5	1	0.3
S15	RF00114	0.56	0.85	1	0.85	1	0.85	1	0.85	1	0.85	1	0.83	1	0.82
SAM	RF00162	0.55	0.72	0.81	0.99	0.67	0.98	0.76	0.91	0.98	1	0.99	0.99	1	0.79
s2m	RF00164	1	1	1	1	1	1	1	0.97	1	0.97	1	0.18	1	0.87
Purine	RF00167	0.66	0.71	1	1	1	1	1	0.99	1	1	1	0.99	1	0.94
Lysine	RF00168	0.82	0.94	1	0.98	1	0.89	1	0.85	1	0.94	1	0.98	1	0.45
Bacteria_small_SRP	RF00169	0.64	0.75	0.99	0.77	0.99	0.75	1	0.27	1	0.67	1	0.41	1	0.39
Cobalamin	RF00174	0.73	0.6	0.87	0.99	0.91	1	1	0.94	1	0.96	1	0.97	0.9	0.9
HIV-1_DIS	RF00175	1	0.99	1	0.99	1	0.99	1	0.99	1	1	1	0.91	1	0.91
SSU_rRNA_bacteria	RF00177	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K10_TLS	RF00207	1	1	1	1	1	1	1	1	0.8	1	1	1	1	1
IRES_Pesti	RF00209	0.95	1	0.91	1	0.91	1	1	1	1	1	0.96	1	1	1
glmS	RF00234	0.69	1	0.97	1	1	1	1	0.89	1	0.89	1	1	1	0.89
Gammaretro_CES	RF00374	0.64	1	0.87	1	0.87	1	1	1	1	1	0.99	1	1	1
ykoK	RF00380	0.7	0.86	1	0.94	1	0.96	1	0.89	1	0.99	1	0.99	0.95	0.99
IRES_Cripavirus	RF00458	0.33	0.14	0.25	0.29	0.25	0.29	1	0.29	1	1	1	0.86	1	1
HIV_FE	RF00480	1	0.98	1	0.98	1	0.98	1	0.98	1	0.99	1	0.99	1	0.97
TCV_H5	RF00500	1	0.8	1	0.8	1	0.8	1	0.8	1	1	1	1	1	1
Glycine	RF00504	0.69	0.59	0.91	0.77	0.87	0.7	0.91	0.52	0.99	0.66	1	0.09	1	0.09
mir-228	RF00843	1	1	1	1	1	1	1	1	1	1	1	1	1	1
mir-689	RF00871	0.5	0.08	0.5	0.08	0.5	0.08	0.83	0.38	0.83	0.38	0.92	0.38	1	0.46
c-di-GMP-I	RF01051	0.81	0.97	1	0.97	0.94	0.94	1	0.96	1	0.98	1	0.76	1	0.69
preQ1-II	RF01054	0.67	0.93	1	0.93	1	0.93	1	0.71	1	0.93	1	0.86	1	0.71
GP_knot1	RF01073	0.96	0.86	0.96	0.71	0.96	0.71	0.91	0.71	0.93	0.86	1	0.43	0.88	0.71
PK-G12rRNA	RF01118	0.65	0.99	0.73	0.99	0.68	0.97	0.99	0.99	1	1	1	1	1	1
HIV-1_SD	RF01380	1	0.05	1	0.05	1	0.05	1	0.05	1	0.77	1	0	1	0

Continued on next page

Table 1 – continued from previous page

Rfam id	Rfam name	A	B	C	D	E	F	G	H	I	J	K	L	M	N
MFR	RF01510	1	0.33	1	0.67	1	0.67	1	0.67	1	1	1	1	0.67	1
AdoCbl-variant	RF01689	1	0.91	1	0.91	1	0.91	1	0.91	1	0.91	1	0.86	1	0.91
crcB	RF01734	0.84	0.36	0.93	0.45	1	0.36	0.88	0.28	0.88	0.32	1	0.03	1	0.03
c-di-GMP-II	RF01786	0.67	0.02	1	0.04	1	0.04	1	0.04	1	0.07	1	0.04	1	0.02
THF	RF01831	0.76	0.96	1	0.96	0.86	0.87	0.7	0.14	0.8	0.24	1	0.24	1	0.16
tRNA-Sec	RF01852	0.89	0.3	0.92	0.3	0.9	0.3	0.53	0.28	0.53	0.28	0.88	0.29	0.92	0.28
Protozoa_SRP	RF01856	0.72	0.39	0.91	0.39	0.95	0.39	1	0.11	1	0.33	1	0.33	1	0.33
Archaea_SRP	RF01857	0.35	0.96	0.82	0.96	0.45	0.96	1	0.87	1	0.96	1	1	1	0.15
group-II-D1D4-1	RF01998	0.38	0.5	0.38	0.46	0.45	0.31	0.62	0.11	0.62	0.06	0.83	0.02	0.5	0.02
group-II-D1D4-3	RF02001	0.3	0.78	1	0.98	1	0.99	1	0.98	1	0.96	0.94	0.94	0.97	0.49
mir-2985-2	RF02095	0.68	0.95	0.84	0.95	0.97	1	1	0.95	1	1	1	1	1	1
IRE_II	RF02253	0.42	0.17	0.42	0.17	0.44	0.17	0.54	0.17	0	0.17	0	0.03	0	0.17
ToxI	RF02519	0.78	0.5	0.56	0.5	0.62	0.5	0.62	0.5	1	1	1	1	1	1

D Diverse Rfam RNA families benchmark set

The Rfam database features following tags to group families: Cis-reg, frameshift_element, IRES, leader, riboswitch, thermoregulator, antisense, antitoxin, CRISPR, lncRNA, miRNA, ribozyme, rRNA, snRNA, snoRNA, CD-box, HACA-box, scaRNA, splicing, Gene, sRNA, tRNA, Intron.

To obtain a representative sample of Rfam families, for each of these tags the alphanumerically first 10 families (if available for that tag) were selected. As some families have multiple tags, the list was filtered to contain each family only once.

The benchmark was conducted in the same manner as for the families with known 3D structure. The plots show different combinations of e-value cutoffs and databasesizes. Without explicitly setting the database size cmsearch uses twice the sequence length (forward/backward strand).

The setting comparable to the one used for the structured dataset is diverse(ev-1e-3,db-1e-9), meaning a cmsearch e-value cutoff of 1e-3 and a databasesize of 10^9 bases in general and 10^6 bases for bacterial and viral RNA families.

The result with comparable settings to the structured dataset has 191 of 192 cases (99%) with at least half of the sequences collected by **RNAlie**n are recognized as belonging to the **Rfam** model. In 170 (89%) families all sequences included by **RNAlie**n are recognized as belonging to the **Rfam** model

In of 163 cases (85%) at least half the sequences in the **Rfam** seed alignment are correctly recognized by the **RNAlie**n model. In 123 of 191 cases (64%) all sequences in the **Rfam** seed alignment are correctly recognized by the **RNAlie**n model.

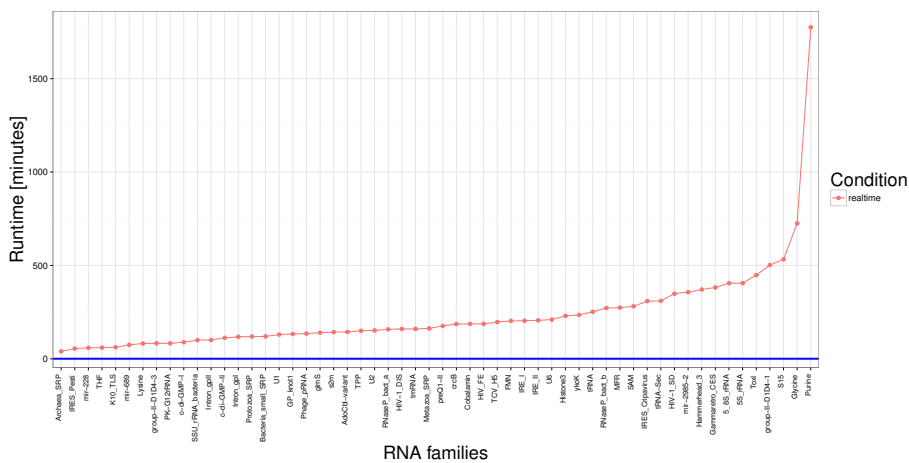


Fig. 7. Alien program runtime in minutes for structured families

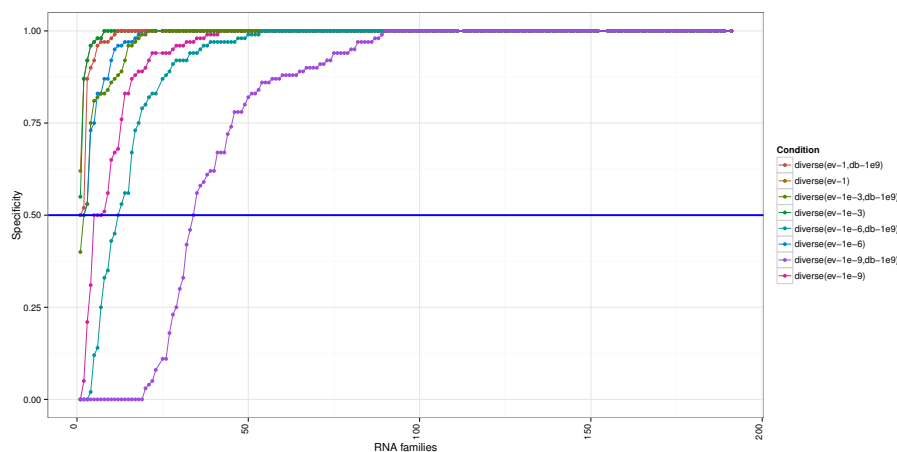


Fig. 8. Specificity of RNAlieN homology search. The plot shows the fraction of homologs predicted by RNAlieN that are recognized by the original Rfam model. The legend indicates the e-evaluate cutoff (ev-) and the database size used. The e-evaluate cutoffs start at 1 and are made stricter in 1e-3 steps up to 1e-9. The result with comparable settings to the structured dataset has 191 of 192 cases (99%) with at least half of the sequences collected by RNAlieN are recognized as belonging to the Rfam model. In 170 (89%) families all sequences included by RNAlieN are recognized as belonging to the Rfam model

Following is the table of families from the Rfam 12.0 used in the as a second benchmark set.

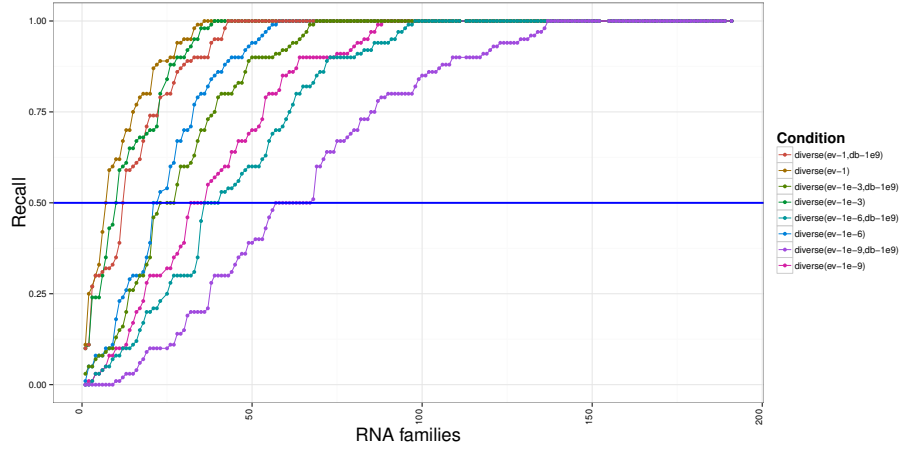


Fig. 9. Recall for 191 RNA families, selected up to 10 for each family tag. To test our method, sRNA Rfam family models were reconstructed by **RNAlien** from a random sequence picked from the family seed sequences. This plot shows how many **Rfam** seed sequences are recognized by the reconstructed **RNAlien** model using the model gathering score (used by **Rfam** to establish full models). In of 163 cases (85%) at least half the sequences in the **Rfam** seed alignment are correctly recognized by the **RNAlien** model. In 123 of 191 cases (64%) all sequences in the **Rfam** seed alignment are correctly recognized by the **RNAlien** model.

Table 2: Diverse RNA families benchmark set. Column names A to H are placeholders for following names: Specificity_value_1 (=A) Sensitivity_value_1 (=B) Specificity_value_1e-3 (=C) Sensitivity_value_1e-3 (=D) Specificity_value_1e-6 (=E) Sensitivity_value_1e-6 (=F) Specificity_value_1e-9 (=G) Sensitivity_value_1e-9 (=H)

Rfam name	Rfam id	A	B	C	D	E	F	G	H
5S_rRNA	RF00001	1	0.88	1	0.75	0.97	0.54	0.58	0.48
5_8S_rRNA	RF00002	1	0.89	1	0.82	1	0.75	1	0.69
U1	RF00003	1	1	1	1	0.98	0.98	0.94	0.94
U2	RF00004	1	1	0.99	0.96	0.92	0.83	0.88	0.75
tRNA	RF00005	1	0.61	0.75	0.47	0.02	0.25	0	0.03
RNaseP_nuc	RF00009	1	0.11	1	0.09	1	0.08	1	0.07
RNaseP_bact_b	RF00011	1	1	1	1	1	1	1	1
U4	RF00015	0.97	0.74	0.96	0.55	0.94	0.12	0.9	0.06
Y_RNA	RF00019	0.52	0.59	0.4	0.33	0.12	0.31	0.05	0.31
U5	RF00020	1	0.9	1	0.61	0.97	0.21	0.78	0.09
U6	RF00026	1	0.94	1	0.8	0.99	0.63	0.88	0.51
PrfA	RF00038	1	1	1	1	1	1	1	1
CopA	RF00042	1	1	1	1	1	1	0.97	0.97
FMN	RF00050	1	1	1	1	0.99	0.99	0.8	0.78

Continued on next page

Table 2 – continued from previous page

Rfam name	Rfam id	A	B	C	D	E	F	G	H
TPP	RF00059	1	0.95	1	0.83	1	0.53	0.67	0.39
U7	RF00066	1	0.71	0.99	0.63	0.79	0.53	0.18	0.39
SNORD29	RF00070	1	0.3	1	0.3	1	0.3	0	0.3
mir-29	RF00074	0.96	1	0.82	0.6	0.56	0.6	0.11	0.4
RNAI	RF00106	1	1	1	1	0.97	1	0.91	0.9
SIB_RNA	RF00113	1	1	1	1	1	1	1	1
snoZ159	RF00160	1	0.3	1	0.1	0.33	0.1	0	0.1
Hammerhead_1	RF00163	1	0.31	1	0.1	0	0.03	0	0.03
Purine	RF00167	1	0.89	1	0.83	1	0.55	1	0.14
SSU_rRNA_bacteria	RF00177	1	1	1	1	1	1	1	1
IRES_Bag1	RF00222	1	1	1	1	1	1	1	1
glmS	RF00234	1	1	1	0.89	1	0.89	1	0.33
ctRNA_pGA1	RF00236	1	1	1	1	1	0.6	1	0.2
RNA-OUT	RF00240	1	1	1	1	1	1	1	1
ctRNA_pT181	RF00242	1	1	1	0.94	1	0.69	1	0.62
IRES_L-myc	RF00261	1	1	1	1	0.82	1	0.23	1
SCARNA18	RF00283	1	1	1	1	1	1	0.86	0.95
SCARNA8	RF00286	1	1	1	1	1	1	1	1
snoR86	RF00303	1	1	1	1	1	1	0.88	1
snoZ157	RF00333	1	1	1	1	0.94	0.9	0.83	0.8
snoR60	RF00339	1	1	0.96	1	0.96	1	0.72	0.9
ydaO-yuaA	RF00379	1	1	1	0.99	1	0.94	0.9	0.84
Antizyme_FSE	RF00381	1	1	1	0.92	0.99	0.62	0.91	0.46
Pox_AX_element	RF00384	1	1	1	1	1	1	1	1
IBV_D-RNA	RF00385	1	1	1	1	1	1	1	0.9
SNORA30	RF00415	1	1	1	0.73	1	0.73	1	0.64
SCARNA24	RF00422	1	1	1	1	1	1	1	0.87
SCARNA15	RF00426	1	1	1	1	1	0.77	0.88	0.64
SCARNA23	RF00427	1	1	1	1	1	1	1	0.94
Hsp90_CRE	RF00433	1	1	1	1	1	1	1	1
ROSE	RF00435	1	1	1	0.46	1	0.15	0	0.15
IRES_HIF1	RF00449	1	1	1	1	1	0.94	0.87	0.88
IRES_mnt	RF00457	1	1	1	1	1	1	1	0.95
HCV_SLVII	RF00468	1	1	1	1	1	1	1	0.91
HCV_SLIV	RF00469	1	1	1	1	1	1	1	0.94
SCARNA6	RF00478	1	1	1	0.94	1	0.94	1	0.94
HIV_FE	RF00480	1	1	1	0.99	1	0.99	0.92	0.86
IRES_Cx43	RF00487	1	1	1	1	0.98	1	0.83	0.93
U1_yeast	RF00488	1	1	1	1	1	1	1	0.8
ctRNA_p42d	RF00489	0.97	1	0.81	0.9	0.53	0.6	0.03	0.2
IRES_Hsp70	RF00495	1	0.86	1	0.86	1	0.86	1	0.86
RNAIII	RF00503	1	1	1	1	0.56	1	0.56	1
Thr_leader	RF00506	1	1	1	1	1	1	0.95	0.96
snosnR64	RF00509	1	1	1	1	0.91	1	0.82	0.9
Leu_leader	RF00512	1	1	1	1	1	1	1	1
Trp_leader	RF00513	1	0.95	1	0.91	0.98	0.59	0.78	0.36
His_leader	RF00514	1	1	1	1	1	0.97	0.99	0.79

Continued on next page

Table 2 – continued from previous page

Rfam name	Rfam id	A	B	C	D	E	F	G	H
PreQ1	RF00522	1	0.74	0.84	0.26	0.14	0.17	0	0.14
Flavivirus_DB	RF00525	1	1	1	1	1	0.92	1	0.68
snoMe28S-G3255	RF00527	1	1	1	0.5	1	0.3	0	0.1
IRES_TrkB	RF00547	1	1	1	1	1	0.94	1	0.94
IRES_c-sis	RF00549	1	1	1	1	1	1	1	1
L13_leader	RF00555	1	0.35	1	0.35	1	0.35	0.95	0.35
L19_leader	RF00556	1	1	1	0.6	0.88	0.2	0	0.2
L20_leader	RF00558	1	0.79	1	0.3	1	0.21	0.94	0.21
L21_leader	RF00559	1	0.87	1	0.74	0.96	0.45	0.46	0.11
SCARNA3	RF00565	1	1	1	1	1	1	1	0.96
SCARNA14	RF00582	1	1	1	1	1	1	1	0.86
CoTC_ribozyme	RF00621	1	1	1	1	1	1	1	0.9
CPEB3_ribozyme	RF00622	1	1	1	1	1	0.92	1	0.67
P1	RF00623	1	1	1	1	1	0.86	1	0.5
P24	RF00629	1	1	1	1	1	1	1	1
MIR169.2	RF00645	1	0.32	1	0.07	1	0.03	1	0.03
MIR168	RF00677	1	1	1	1	0.97	0.9	0.87	0.6
MIR162.2	RF00742	1	1	1	0.9	0.75	0.1	0.25	0.1
mir-342	RF00760	1	1	1	1	1	1	1	1
mir-541	RF00777	1	1	1	0.9	0.92	0.9	0.62	0.9
mir-1255	RF00994	0.99	0.9	0.86	0.9	0.45	0.9	0.11	0.1
WLE3	RF01046	1	1	1	0.7	1	0.7	1	0.6
Sacc_telomerase	RF01050	1	1	1	1	1	1	1	1
preQ1-II	RF01054	1	1	1	0.93	1	0.71	1	0.64
MOCO_RNA_motif	RF01055	1	0.33	1	0.13	1	0.07	1	0.02
RF_site2	RF01076	1	1	0.83	1	0.67	1	0.67	1
RF_site3	RF01079	1	1	1	1	1	1	1	0.5
RF_site5	RF01093	1	1	1	1	1	0.58	0.9	0.5
RF_site9	RF01098	1	1	1	1	1	1	1	1
PK-G12rRNA	RF01118	1	1	1	1	1	1	1	1
snoZ30a	RF01196	1	1	1	1	1	1	1	1
snoR103	RF01213	0.87	1	0.87	1	0.87	0.82	0.87	0.73
snoR442	RF01232	1	1	1	1	0.25	0.7	0	0.1
snR161	RF01237	1	1	1	0.9	1	0.9	1	0.5
snR36	RF01242	1	1	1	1	1	1	1	1
snR8	RF01248	1	1	1	1	1	0.91	1	0.91
snR190	RF01249	1	1	1	1	1	0.8	1	0.8
snR5	RF01252	1	1	1	1	1	0.82	1	0.82
snR35	RF01255	1	1	1	1	1	1	1	1
snR191	RF01263	1	1	1	1	1	1	1	1
SCARNA2	RF01268	1	0.95	1	0.95	1	0.95	1	0.95
snoR2	RF01292	1	1	1	1	1	1	1	1
SCARNA7	RF01295	1	1	1	1	1	0.94	1	0.94
AHBV_epsilon	RF01313	1	1	1	1	1	1	0.88	1
CRISPR-DR2	RF01315	1	0.74	1	0.05	0	0.05	0	0
CRISPR-DR3	RF01316	0.5	0.1	0.5	0.05	0	0	0	0
CRISPR-DR5	RF01318	1	1	1	0.08	1	0.08	0	0

Continued on next page

Table 2 – continued from previous page

Rfam name	Rfam id	A	B	C	D	E	F	G	H
CRISPR-DR7	RF01320	1	0.9	1	0.2	1	0.1	0	0
CRISPR-DR35	RF01345	1	1	1	1	1	1	1	0
CRISPR-DR53	RF01366	1	1	1	1	1	1	1	0
CRISPR-DR60	RF01373	1	1	1	1	1	0.5	0	0.5
CRISPR-DR61	RF01374	1	1	0.83	1	0.83	1	0	0.5
CRISPR-DR65	RF01378	1	1	1	1	1	1	0	0
isrA	RF01385	1	1	1	1	0.97	1	0.97	1
istR	RF01400	1	1	1	1	1	1	0.97	1
NrrF	RF01416	1	1	1	1	1	1	1	1
IsrR	RF01419	1	0.98	1	0.97	1	0.91	1	0.88
VrrA	RF01456	1	1	1	1	0.95	1	0.84	1
Afu_300	RF01509	1	1	1	1	1	1	0.61	0.5
MFR	RF01510	1	1	1	1	1	1	1	0.67
Afu_309	RF01512	1	1	1	1	1	1	1	1
Dictyostelium_SRP	RF01570	1	1	1	1	1	1	1	1
RNase_P	RF01577	1	1	1	1	1	1	1	1
AdoCbl-variant	RF01689	1	0.62	1	0.03	1	0.01	1	0.01
Lnt	RF01711	1	0.9	1	0.8	1	0.3	0	0.3
cspA	RF01766	1	1	1	1	1	1	1	1
SMK_box_riboswitch	RF01767	1	0.6	1	0.08	1	0.04	1	0.04
rnk_leader	RF01771	0.97	1	0.97	1	0.97	0.85	0.97	0.85
RatA	RF01776	1	1	0.88	1	0.35	0.56	0.04	0.5
blv_FSE	RF01785	1	1	1	1	1	0	0	0
FourU	RF01795	1	1	1	1	1	1	0.94	1
fstAT	RF01797	1	1	1	1	0.94	1	0.94	0.73
HSUR	RF01802	1	0.5	1	0.5	1	0.5	1	0.5
Lambda_thermo	RF01804	1	1	1	1	1	1	1	1
GIR1	RF01807	1	1	0.89	0.92	0.89	0.92	0.89	0.92
MicX	RF01808	1	1	1	1	1	1	1	1
symR	RF01809	1	1	1	1	1	1	1	1
PtaRNA1	RF01811	1	1	1	1	1	1	1	0.75
rdlD	RF01813	1	1	1	1	1	1	1	0.98
ROSE_2	RF01832	1	1	1	1	0.99	1	0.94	1
HIV_FS2	RF01835	1	1	1	1	1	1	1	0.79
ovine_lenti_FSE	RF01840	1	1	1	1	1	1	1	0.93
veev_FSE	RF01841	1	1	1	1	0.5	0.9	0.5	0.7
alpha_tmRNA	RF01849	1	1	1	1	1	1	0.98	1
tRNA-Sec	RF01852	0.9	0.32	0.53	0.28	0.43	0.28	0.08	0.28
MIAT_exon1	RF01874	1	1	1	1	1	1	0.98	0.9
MIAT_exon5_2	RF01876	1	1	1	1	1	1	1	1
HSR-omega_2	RF01886	1	1	1	1	1	1	0.86	1
mir-2241	RF01899	1	1	1	0.5	1	0.5	1	0.5
mir-284	RF01901	1	1	1	1	1	1	1	1
HEARO	RF02033	1	0.27	1	0.16	1	0.05	1	0.01
STnc630	RF02052	1	1	1	1	1	1	1	1
STnc370	RF02064	1	1	1	1	1	0.8	1	0.8
STnc180	RF02079	1	0.8	1	0.5	1	0.3	1	0.3

Continued on next page

Table 2 – continued from previous page

Rfam name	Rfam id	A	B	C	D	E	F	G	H
OrzO-P	RF02083	1	1	1	1	1	1	1	0.43
Yar_1	RF02085	1	0.9	1	0.8	1	0.5	1	0.3
tfoR	RF02100	1	1	1	1	1	1	1	1
IS009	RF02111	1	1	1	1	0.97	1	0.74	0.73
FAM13A-AS1.1	RF02114	0.92	1	0.92	0.8	0.92	0.3	0.92	0.2
FAM13A-AS1.2	RF02115	1	0.8	1	0.6	1	0.3	1	0.3
MEG8.3	RF02147	1	1	1	1	0.92	0.6	0.42	0.4
PVT1_4	RF02167	1	1	1	0.9	1	0.5	1	0.4
HPnc0260	RF02194	1	0.39	1	0.26	1	0.23	1	0.19
WT1-AS_1	RF02203	1	1	1	1	1	1	0.9	0.8
sX5	RF02224	1	1	1	1	1	1	0.59	0.8
sX11	RF02230	1	1	1	1	1	1	1	1
Six3os1_3	RF02248	1	1	1	1	1	0.9	0.89	0.8
Hammerhead_II	RF02276	1	0.83	1	0.79	0.73	0.54	0	0
Hammerhead_HH10	RF02277	1	1	1	1	1	1	1	1
hsp17	RF02358	1	0.67	1	0.67	0.83	0.67	0.33	0.67
PyrG_leader	RF02371	1	1	1	0.7	0.8	0.2	0.3	0.2
PyrD_leader	RF02373	1	0.59	1	0.15	1	0.11	1	0.11
Ms_AS-8	RF02466	1	1	1	1	1	1	0.78	0.8
GLRNase_MRP	RF02472	1	1	1	1	1	1	1	1
GLU1	RF02491	1	1	1	1	1	1	1	1
GLU2	RF02492	1	1	1	1	1	1	1	1
GLU4	RF02493	1	1	1	1	1	1	1	1
GLU6	RF02494	1	1	1	1	1	1	1	1
ohsC_RNA	RF02495	1	1	1	1	1	0.97	1	0.97
mir-2494	RF02518	1	1	1	0.9	1	0.9	1	0.7
ToxI	RF02519	1	1	1	1	1	1	0.62	0.5
ROSE_3	RF02523	1	1	1	1	1	1	1	0.88
NRF2_IRES	RF02531	0.98	1	0.98	1	0.97	0.95	0.86	0.9
MNV_3UTR	RF02532	1	1	1	1	1	1	1	1
ODC_IRES	RF02535	1	1	1	1	1	1	0.97	0.85
mt-tmRNA	RF02544	1	1	1	0.91	1	0.82	0.67	0.36

E Negative control set

We used coding sequences, ancestral repeats, untranslated regions (UTRs) and random sequences to perform a negative control. According to the procedure for structured and diverse RNA families the sequences of the negative control set were used as a input sequence for **RNAlien**. Taxonomic start points for the construction were set as below using taxids from NCBI taxonomy [2]. The results were summarized for each subset individually.

E.1 Random sequences

A test with 300 different 100 nucleotides long random sequences was performed. 100 Sequences each were used in *Escherichia coli*, *Homo sapiens* and *Sulfolobus solfataricus*. The sequences were created with a inhouse *randseq* program, source code will be provided on request by Ivo L. Hofacker (ivo@tbi.univie.ac.at).

E.2 Ancestral repeats

All 62 entries tagged with ancestral repeat from the **Dfam** [3] database were used with *Homo sapiens* as starting point for RNAlien, if the repeat was present there. The exceptions are the following list of pairs, with the first element containing the family name and the second the taxonomic start point: (Charlie12_Rodent, *Mus musculus*), (DNA9TA1_DR, *Danio rerio*), (L2-1_DR, *Danio rerio*), (Jockey2, *Drosophila melanogaster*), (DIVER2_I, *Drosophila melanogaster*)

E.3 Coding sequences

50 Protein coding sequences were checked for *Escherichia coli*, *Sulfolobus solfataricus* and *Homo sapiens*. *Escherichia coli* sequences are the first 50 annotated CDS sequences from regulonDB 9.0 [4] (http://regulondb.ccg.unam.mx/menu/download/datasets/files/Gene_sequence.txt) . *Sulfolobus solfataricus* sequences are retrieved from the reference genbank [5] assembly for *Sulfolobus solfataricus* *GCF_000007005.1_ASM700v1*. *Homo sapiens* sequences are from Ensemble [6] (Release 84, GRCh38.p5), chromosome2.

E.4 UTR regions

50 3-prime and 5-prime untranslated regions from *E.coli* and *Homo sapiens* were checked. *Escherichia coli* sequences are from regulonDB version 9.0 [4] (http://regulondb.ccg.unam.mx/menu/download/datasets/files/UTR_5_3_sequence.txt), *Homo sapiens* sequences are from Ensemble [6] (Release 84, GRCh38.p5), chromosome2. For *Sulfolobus* we could not find a UTR dataset.

Table 3: Table for negative control set construction results. Shown are selected result fields of RNAs, RNAcode and cmstat. Column names A to Y are placeholders for following names: Name(=A), alienFastaNumber(=B), meanPairwiseIdentity(=C), shannonEntropy(=D), gcContent(=E), meanSingleSequenceMFE(=F), consensusMFE(=G), energyContribution(=H), covarianceContribution(=I), combinationsPair(=J), meanZScore(=K), SCI(=L), svmDecisionValue(=M), svmRNAClassProbability(=N), prediction(=O), RNAcodeLowestp-value(=P), RNAcodeClassification(=Q), statSequenceNumber(=R), statEffectiveSequences(=S), statConsensusLength(=T), statW(=U), statBasepairs(=V), statBifurcations(=W), relativeEntropyCM(=X), relativeEntropyHMM(=Y)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
hs.random1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	28	1	0.591	0.318
hs.random2	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.61	100	118	34	1	0.59	0.266
hs.random3	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	24	2	0.589	0.369
hs.random4	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	26	3	0.59	0.34
hs.random5	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	2	0.591	0.335
hs.random6	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	4	0.592	0.335
hs.random7	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	117	27	2	0.589	0.335
hs.random8	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	119	28	2	0.59	0.319
hs.random9	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	29	0	0.589	0.322
hs.random10	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	132	26	2	0.59	0.348
hs.random11	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	117	33	1	0.589	0.276
hs.random12	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	28	1	0.59	0.321
hs.random13	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	20	1	0.589	0.401
hs.random14	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.62	100	118	33	2	0.589	0.278
hs.random15	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.85	100	117	19	2	0.591	0.414
hs.random16	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	23	0	0.588	0.371
hs.random17	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	1	0.59	0.336
hs.random18	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	26	1	0.589	0.343
hs.random19	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	26	2	0.591	0.34
hs.random20	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	29	2	0.591	0.328
hs.random21	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	118	33	2	0.59	0.282
hs.random22	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	117	31	2	0.589	0.299
hs.random23	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	27	1	0.59	0.331
hs.random24	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	31	1	0.591	0.291
hs.random25	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	30	3	0.589	0.303
hs.random26	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	31	1	0.59	0.287
hs.random27	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	117	30	1	0.59	0.306
hs.random28	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	119	28	2	0.589	0.326
hs.random29	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	117	30	1	0.591	0.308
hs.random30	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.62	100	117	35	1	0.59	0.249
hs.random31	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	27	1	0.589	0.334
hs.random32	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.94	100	118	18	1	0.592	0.432
hs.random33	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.87	100	118	22	0	0.59	0.39
hs.random34	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	119	32	1	0.59	0.288
hs.random35	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	28	2	0.59	0.318
hs.random36	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	26	2	0.591	0.341
hs.random37	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	25	2	0.589	0.35
hs.random38	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	119	32	2	0.59	0.281
hs.random39	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	24	2	0.59	0.365
hs.random40	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	117	24	1	0.591	0.369
hs.random41	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	26	2	0.589	0.345
hs.random42	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	25	1	0.591	0.356
hs.random43	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	117	25	2	0.59	0.352
hs.random44	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	2	0.59	0.33
hs.random45	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	118	30	1	0.59	0.305
hs.random46	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	23	2	0.591	0.371
hs.random47	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	22	1	0.588	0.376
hs.random48	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.59	100	118	34	1	0.59	0.259
hs.random49	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	118	30	1	0.589	0.299
hs.random50	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	117	26	1	0.59	0.347
hs.random51	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	29	0	0.59	0.313
hs.random52	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	31	2	0.589	0.289
hs.random53	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	26	1	0.589	0.351
hs.random54	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	28	2	0.589	0.328
hs.random55	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.64	100	118	31	1	0.591	0.289
hs.random56	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	25	0	0.591	0.354
hs.random57	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	117	31	1	0.591	0.298
hs.random58	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	25	1	0.592	0.358
hs.random59	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	26	3	0.591	0.346
hs.random60	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	29	1	0.59	0.319
hs.random61	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	31	0	0.591	0.287
hs.random62	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	28	0	0.59	0.322
hs.random63	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	119	26	1	0.59	0.342
hs.random64	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.83	100	118	25	1	0.591	0.356
hs.random65	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.91	100	117	21	0	0.589	0.395
hs.random66	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	25	1	0.589	0.363
hs.random67	1	-	-	-	-	-	-	-	-	-														

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
hs_random80	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.83	100	118	23	2	0.589	0.379
hs_random81	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	24	0	0.591	0.363
hs_random82	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	23	2	0.589	0.364
hs_random83	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	117	32	1	0.591	0.284
hs_random84	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	27	2	0.59	0.337
hs_random85	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	31	1	0.591	0.304
hs_random86	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	30	2	0.59	0.308
hs_random87	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	24	1	0.588	0.372
hs_random88	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	28	2	0.59	0.326
hs_random89	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.89	100	118	21	2	0.589	0.398
hs_random90	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	26	1	0.59	0.342
hs_random91	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	132	25	2	0.59	0.348
hs_random92	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	27	1	0.589	0.339
hs_random93	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	23	1	0.589	0.378
hs_random94	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	25	0	0.589	0.348
hs_random95	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	31	1	0.59	0.298
hs_random96	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	24	2	0.59	0.362
hs_random97	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	117	33	2	0.591	0.274
hs_random98	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	28	1	0.59	0.324
hs_random99	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	132	28	2	0.591	0.332
hs_random100	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	119	29	2	0.591	0.316
ec_random101	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	30	1	0.592	0.306
ec_random102	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	23	3	0.589	0.368
ec_random103	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	1	0.589	0.333
ec_random104	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	2	0.589	0.335
ec_random105	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	30	3	0.59	0.307
ec_random106	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	27	1	0.589	0.329
ec_random107	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	21	2	0.592	0.4
ec_random108	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	119	27	3	0.589	0.331
ec_random109	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	27	2	0.591	0.336
ec_random110	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	117	27	0	0.591	0.339
ec_random111	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	117	29	0	0.591	0.329
ec_random112	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	29	0	0.589	0.313
ec_random113	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	32	0	0.591	0.29
ec_random114	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	25	2	0.591	0.369
ec_random115	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	31	0	0.59	0.303
ec_random116	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	24	2	0.591	0.366
ec_random117	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	119	28	1	0.59	0.319
ec_random118	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	27	0	0.589	0.334
ec_random119	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	26	2	0.59	0.344
ec_random120	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	23	1	0.591	0.374
ec_random121	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	119	30	1	0.59	0.306
ec_random122	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	28	1	0.591	0.321
ec_random123	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	29	0	0.59	0.316
ec_random124	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	119	29	2	0.589	0.31
ec_random125	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.64	100	118	31	0	0.59	0.305
ec_random126	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	29	2	0.589	0.319
ec_random127	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	132	28	3	0.591	0.327
ec_random128	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	26	0	0.591	0.346
ec_random129	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	28	1	0.589	0.328
ec_random130	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	27	1	0.592	0.333
ec_random131	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	27	2	0.589	0.331
ec_random132	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	24	2	0.59	0.372
ec_random133	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	117	34	0	0.591	0.273
ec_random134	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	29	2	0.589	0.323
ec_random135	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	25	2	0.589	0.359
ec_random136	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	117	24	0	0.59	0.363
ec_random137	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	25	2	0.589	0.348
ec_random138	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	1	0.591	0.34
ec_random139	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	25	0	0.591	0.358
ec_random140	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	118	30	2	0.591	0.309
ec_random141	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	25	1	0.588	0.355
ec_random142	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	28	1	0.59	0.322
ec_random143	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	1	0.589	0.345
ec_random144	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.81	100	118	25	1	0.588	0.348
ec_random145	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	25	1	0.59	0.355
ec_random146	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	24	3	0.591	0.36
ec_random147	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.98	100	118	18	1	0.591	0.429
ec_random148	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.89	100	118	18	1	0.591	0.418
ec_random149	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.64	100	118	32	2	0.591	0.289
ec_random150	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	29	1	0.591	0.318
ec_random151	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	30	1	0.589	0.313
ec_random152	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	119	29	2	0.589	0.312
ec_random153	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	24	2	0.589	0.365
ec_random154	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	118	32	3	0.591	0.288
ec_random155	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	28	2	0.59	0.328
ec_random156	1	-	-																					

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ec_random168	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	29	2	0.592	0.319
ec_random169	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	23	3	0.59	0.377
ec_random170	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	30	1	0.59	0.301
ec_random171	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	30	1	0.589	0.3
ec_random172	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	116	27	0	0.59	0.331
ec_random173	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	32	0	0.591	0.294
ec_random174	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	29	2	0.591	0.316
ec_random175	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	30	1	0.589	0.307
ec_random176	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	27	1	0.589	0.333
ec_random177	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	29	1	0.591	0.317
ec_random178	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	119	26	1	0.589	0.34
ec_random179	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	119	26	1	0.591	0.345
ec_random180	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	25	0	0.589	0.356
ec_random181	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	29	1	0.589	0.325
ec_random182	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	26	0	0.589	0.349
ec_random183	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	23	2	0.589	0.369
ec_random184	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.89	100	117	24	1	0.589	0.375
ec_random185	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	27	4	0.591	0.335
ec_random186	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	119	21	2	0.59	0.388
ec_random187	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.61	100	118	33	1	0.589	0.274
ec_random188	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.61	100	118	33	1	0.589	0.269
ec_random189	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	32	3	0.589	0.278
ec_random190	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	28	1	0.59	0.329
ec_random191	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	24	2	0.59	0.369
ec_random192	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	29	1	0.591	0.321
ec_random193	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	25	2	0.59	0.351
ec_random194	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.85	100	118	22	1	0.59	0.388
ec_random195	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	26	2	0.589	0.349
ec_random196	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	117	30	0	0.59	0.297
ec_random197	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	117	28	1	0.59	0.332
ec_random198	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	132	27	2	0.591	0.345
ec_random199	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.87	100	118	23	1	0.592	0.388
ec_random200	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	119	27	3	0.589	0.334
ss_random201	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	32	0	0.591	0.294
ss_random202	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	30	0	0.591	0.302
ss_random203	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	131	28	2	0.59	0.324
ss_random204	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.62	100	132	33	2	0.591	0.279
ss_random205	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	119	32	1	0.591	0.282
ss_random206	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	119	27	2	0.589	0.329
ss_random207	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	119	28	1	0.591	0.326
ss_random208	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	1	0.59	0.336
ss_random209	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	3	0.59	0.334
ss_random210	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	30	2	0.591	0.313
ss_random211	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	25	1	0.589	0.358
ss_random212	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	27	4	0.591	0.328
ss_random213	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	25	2	0.59	0.355
ss_random214	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	119	27	3	0.589	0.343
ss_random215	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.85	100	117	23	1	0.588	0.374
ss_random216	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.83	100	118	25	1	0.59	0.356
ss_random217	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	24	2	0.588	0.362
ss_random218	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	2	0.591	0.335
ss_random219	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.81	100	118	21	1	0.591	0.397
ss_random220	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	100	118	27	1	0.59	0.336
ss_random221	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	29	1	0.591	0.314
ss_random222	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	23	1	0.59	0.369
ss_random223	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	117	34	1	0.591	0.268
ss_random224	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.86	100	118	24	1	0.592	0.376
ss_random225	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	24	1	0.59	0.362
ss_random226	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	27	1	0.59	0.34
ss_random227	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	21	1	0.59	0.389
ss_random228	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	117	26	0	0.591	0.341
ss_random229	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.64	100	118	32	1	0.589	0.279
ss_random230	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	27	2	0.59	0.344
ss_random231	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	26	2	0.588	0.342
ss_random232	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	26	1	0.591	0.339
ss_random233	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	33	0	0.59	0.293
ss_random234	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	25	2	0.589	0.355
ss_random235	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	117	29	1	0.591	0.319
ss_random236	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.6	100	118	34	1	0.59	0.261
ss_random237	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.87	100	118	18	1	0.588	0.421
ss_random238	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	117	31	2	0.59	0.291
ss_random239	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.64	100	118	32	2	0.589	0.28
ss_random240	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.61	100	118	32	2	0.591	0.284
ss_random241	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.85	100	118	22	0	0.591	0.381
ss_random242	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	25	2	0.59	0.359
ss_random243	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	31	1	0.59	0.293
ss_random244	1																							

Table 3 – continued from previous page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ss_random256	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	117	29	2	0.59	0.324
ss_random257	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	24	1	0.59	0.366
ss_random258	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	119	25	2	0.592	0.355
ss_random259	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	30	1	0.589	0.306
ss_random260	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	32	2	0.591	0.291
ss_random261	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	119	25	1	0.589	0.351
ss_random262	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	30	1	0.591	0.307
ss_random263	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	118	27	1	0.591	0.349
ss_random264	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.8	100	119	23	2	0.591	0.372
ss_random265	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.62	100	118	32	0	0.591	0.28
ss_random266	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	30	2	0.59	0.308
ss_random267	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	27	1	0.589	0.331
ss_random268	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	29	1	0.589	0.314
ss_random269	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	29	1	0.59	0.323
ss_random270	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	119	24	3	0.591	0.36
ss_random271	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	23	2	0.59	0.372
ss_random272	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.62	100	118	32	2	0.591	0.283
ss_random273	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.69	100	119	28	2	0.591	0.323
ss_random274	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	26	0	0.591	0.345
ss_random275	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.82	100	118	22	2	0.59	0.381
ss_random276	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	118	28	2	0.591	0.327
ss_random277	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	26	1	0.592	0.354
ss_random278	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	29	1	0.59	0.316
ss_random279	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.67	100	118	31	2	0.591	0.295
ss_random280	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	31	1	0.589	0.29
ss_random281	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.65	100	118	28	2	0.589	0.321
ss_random282	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	25	1	0.591	0.351
ss_random283	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	31	1	0.591	0.286
ss_random284	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.78	100	118	25	1	0.59	0.357
ss_random285	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.7	100	118	29	2	0.592	0.317
ss_random286	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	117	33	1	0.591	0.277
ss_random287	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	28	1	0.591	0.322
ss_random288	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.73	100	118	29	1	0.591	0.315
ss_random289	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	100	118	25	1	0.589	0.354
ss_random290	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	119	28	2	0.59	0.328
ss_random291	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	118	31	2	0.59	0.289
ss_random292	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.79	100	118	25	2	0.589	0.353
ss_random293	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.86	100	118	21	2	0.589	0.391
ss_random294	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.71	100	118	28	1	0.588	0.316
ss_random295	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.66	100	118	33	1	0.59	0.275
ss_random296	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	23	1	0.588	0.37
ss_random297	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.76	100	118	25	2	0.59	0.356
ss_random298	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.74	100	131	25	1	0.589	0.347
ss_random299	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.77	100	118	27	3	0.591	0.333
ss_random300	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.68	100	118	29	2	0.591	0.311
ancestral301	4	70.78	0.38461	0.44033	-130.03	-74.85	-74.01	-0.84	1.24	0.11	0.58	-2.35	0.00001	OTHER	0.472	OTHER	0.472	4	0.94	429	567	125	6	0.59	0.325
ancestral302	36	83.43	0.32333	0.40133	-227.71	-159.55	-158.33	-1.22	1.26	-0.92	0.7	-0.61	0.159931	OTHER	0.00002425	OTHER	0.00002425	36	2.87	743	2198	137	17	0.59	0.466
ancestral303	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.72	1419	1444	363	28	0.59	0.343
ancestral304	24	79.36	0.38635	0.34247	-141.15	-79.86	-80.67	0.81	1.33	-0.86	0.57	-1.12	0.04612	OTHER	0.453	OTHER	0.453	24	3.44	560	798	122	8	0.59	0.445
ancestral305	21	66.08	0.63164	0.33584	-134.74	-21.02	-22.72	1.69	1.61	0.86	0.16	-4.4	0	OTHER	0.279	OTHER	0.279	21	3.86	613	716	99	10	0.59	0.489
ancestral306	48	74.65	0.4543	0.40367	-134.35	-64.28	-67.74	3.46	1.33	-0.46	0.48	-1.85	0.006953	OTHER	0.174	OTHER	0.174	48	3.8	504	689	89	9	0.59	0.477
ancestral307	26	66.88	0.56316	0.36765	-147.39	-54.05	-52.73	-1.32	1.5	-0.08	0.37	-2.47	0.000005	OTHER	0.88	OTHER	0.88	26	4.56	593	1112	74	6	0.59	0.514
ancestral308	27	55.36	0.72936	0.36114	-179.43	-25.97	-28.09	2.12	1.72	-0.02	0.14	-3.09	0	OTHER	0.193	OTHER	0.193	27	4.07	703	2471	113	12	0.59	0.487
ancestral309	28	81.68	0.33225	0.34702	-140.49	-71.93	-75.53	3.6	1.2	0.81	0.51	-3.87	0	OTHER	0.268	OTHER	0.268	28	4.64	619	968	86	9	0.59	0.505
ancestral310	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ancestral311	3	97.1	0.03993	0.33497	-9.67	-8.9	-9.23	0.33	1	0.8	0.92	-2.97	0	OTHER	0.234	OTHER	0.234	3	1.46	69	86	17	1	0.814	0.611
ancestral312	66	76.09	0.44929	0.28298	-36.06	-7.49	-6.3	-1.19	1.44	0.75	0.21	-4.8	0	OTHER	0.198	OTHER	0.198	66	3.98	226	256	41	2	0.59	0.473
ancestral313	30	64.14	0.67586	0.35367	-126.18	-22.65	-25.78	3.13	1.55	0.65	0.18	-3.88	0	OTHER	0.825	OTHER	0.825	30	4.65	556	695	80	7	0.59	0.501
ancestral314	4	54.65	0.71539	0.35137	-159.78	-31.56	-24.5	-7.06	1.81	-0.17	0.2	-2.66	0.000002	OTHER	0.093	OTHER	0.093	4	1.81	687	882	157	13	0.589	0.409
ancestral315	38	61.23	0.71454	0.3754	-154.57	-30.28	-29.08	-1.2	1.57	-0.85	0.2	-1.84	0.007176	OTHER	0.000695	PROTEIN	0.000695	38	4.65	654	1148	84	8	0.59	0.512
ancestral316	49	61.73	0.7012	0.39975	-138.36	-32.35	-35.58	3.23	1.62	-0.22	0.23	-2.49	0.000005	OTHER	0.068	OTHER	0.068	49	4.9	562	637	69	5	0.59	0.52
ancestral317	43	67.43	0.61204	0.35571	-107.73	-20.28	-18.5	-1.79	1.56	0.72	0.19	-4.18	0	OTHER	0.375	OTHER	0.375	43	5.1	480	545	77	3	0.59	0.496
ancestral318	42	57.09	0.79051	0.37492	-130.29	-24.33	-22.88	-1.45	1.73	0.45	0.19	-3.14	0	OTHER	0.889	OTHER	0.889	42	6.51	496	1324	60	7	0.59	0.519
ancestral319	44	69.88	0.55626	0.33964	-93.68	-25.06	-25.29	0.23	1.49	0.44	0.27	-3.68	0	OTHER	0.819	OTHER	0.819	44	4.68	427	519	80	7	0.59	0.475
ancestral320	22	60.07	0.73414	0.37267	-145.46	-25.14	-23.56	-1.58	1.73	0.28	0.17	-3.27	0	OTHER	0.294	OTHER	0.294	22	3.99	553	1045	102	7	0.59	0.472
ancestral321	31	64.12	0.63227	0.3663	-144.59	-33.73	-31.12	-2.61	1.7	0.4	0.23	-3.52	0	OTHER	0.249	OTHER	0.249</								

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ancestral344	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral345	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral346	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral347	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral348	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral349	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral350	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral351	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral352	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral353	3	86.34	0.18632	0.33011	-7.2	-1.84	-1.07	-0.77	1.4	0.34	0.26	-5.16	0	OTHER	0.639	OTHER	3	1.67	67	83	18	1	0.589	0.346
ancestral354	3	84.42	0.21686	0.26903	-9.57	-6.04	-5.27	-0.77	1.29	0.01	0.63	-2.73	0.000001	OTHER	0.778	OTHER	3	1.26	92	109	24	1	0.621	0.397
ancestral355	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral356	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral357	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral358	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral359	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral360	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral361	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ancestral362	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds363	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds364	293	83.19	0.29658	0.52882	-170.7	-110.88	-111.09	0.21	1.29	-0.21	0.65	-1.96	0.005212	OTHER	3.286E-14	PROTEIN	293	3.11	488	732	90	8	0.59	0.474
hs_cds365	336	79.55	0.35368	0.40477	-108.24	-51.99	-50.84	-1.15	1.29	0.25	0.48	-3.23	0	OTHER	0	PROTEIN	336	2.22	425	447	94	9	0.59	0.431
hs_cds366	245	80.6	0.35557	0.46077	-39.77	-16.89	-16.87	-0.02	1.46	0.49	0.42	-3.84	0	OTHER	2.398E-14	PROTEIN	245	2.15	178	197	36	1	0.59	0.45
hs_cds367	2	98.15	0.01846	0.62158	-979.32	-964.1	-965.85	1.75	1.01	0.86	0.98	-2.82	0.000001	OTHER	-	-	2	0.55	2221	2246	756	37	0.589	0.258
hs_cds368	8	95.49	0.0871	0.55072	-454.53	-408.27	-411.37	3.1	1.09	-1.93	0.9	0.58	0.820264	RNA	3.331E-16	PROTEIN	8	0.66	1165	1194	376	19	0.59	0.282
hs_cds369	258	79.46	0.3491	0.53547	-156.74	-96.35	-98.45	2.1	1.27	-0.28	0.61	-1.84	0.007127	OTHER	4.564E-08	PROTEIN	258	2.37	437	815	82	7	0.59	0.455
hs_cds370	3	97.59	0.03324	0.71897	-516.07	-487.85	-489.53	1.69	1.03	0.59	0.95	-2.59	0.000003	OTHER	0.02	PROTEIN	3	0.52	967	992	331	18	0.589	0.258
hs_cds371	290	83.01	0.29677	0.48942	-117.74	-75.33	-74.12	-1.21	1.28	-0.4	0.64	-1.76	0.008723	OTHER	5.706E-10	PROTEIN	290	2.49	355	477	91	5	0.59	0.418
hs_cds372	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds373	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds374	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds375	2	98.25	0.01748	0.44002	-315.8	-306.05	-311.05	5	1.01	0.26	0.97	-2.15	0.003076	OTHER	-	-	2	0.65	1030	1055	335	17	0.589	0.274
hs_cds376	11	90.74	0.17916	0.53033	-613.79	-502.23	-504.74	2.52	1.11	-0.69	0.82	-0.96	0.068659	OTHER	1.744E-12	PROTEIN	11	0.71	1619	1647	535	24	0.589	0.278
hs_cds377	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds378	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds379	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds380	248	78.79	0.38016	0.50804	-138.16	-68.73	-67.19	-1.54	1.35	-0.64	0.5	-1.84	0.007077	OTHER	1.11E-16	PROTEIN	248	2.49	389	478	90	5	0.59	0.428
hs_cds381	337	84.03	0.28135	0.42118	-117.79	-76.14	-74.9	-1.23	1.28	0.35	0.65	-2.76	0.000001	OTHER	7.772E-16	PROTEIN	337	3.25	443	468	99	11	0.59	0.45
hs_cds382	250	86.15	0.25673	0.59612	-233.35	-148.81	-151.19	2.38	1.27	0.19	0.64	-2.72	0.000001	OTHER	0	PROTEIN	250	1.67	599	646	152	13	0.589	0.395
hs_cds383	304	81.74	0.3406	0.41272	-25.98	-10.99	-11.22	0.23	1.32	0.63	0.42	-4.08	0	OTHER	3.524E-11	PROTEIN	304	1.74	149	168	27	3	0.59	0.45
hs_cds384	66	82.07	0.34165	0.54476	-121.59	-68.96	-69.97	1.01	1.33	0.28	0.57	-2.83	0.000001	OTHER	6.505E-13	PROTEIN	66	1.16	343	365	105	6	0.59	0.334
hs_cds385	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds386	323	80.65	0.33629	0.47296	-102.59	-65.05	-66.2	1.15	1.2	0.2	0.63	-2.41	0.000007	OTHER	2.897E-13	PROTEIN	323	1.84	327	421	87	6	0.59	0.395
hs_cds387	19	76.4	0.43514	0.50991	-450.99	-240.95	-236.93	-4.02	1.37	-0.66	0.53	-1.39	0.023333	OTHER	0	PROTEIN	19	1.04	1260	1320	392	26	0.591	0.32
hs_cds388	272	79.58	0.34846	0.51643	-111.64	-62.58	-64.67	2.09	1.15	0.43	0.56	-3.04	0	OTHER	3.14E-11	PROTEIN	272	1.68	332	558	96	5	0.591	0.361
hs_cds389	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds390	224	73.86	0.44218	0.44178	-174.92	-67.47	-66.88	-0.59	1.37	0.48	0.39	-3.6	0	OTHER	0	PROTEIN	224	2.05	583	1332	112	10	0.59	0.451
hs_cds391	2	98.25	0.01748	0.44002	-315.8	-306.05	-311.05	5	1.01	0.17	0.97	-2.04	0.004121	OTHER	-	-	2	0.65	1030	1055	335	17	0.589	0.274
hs_cds392	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds393	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds394	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds395	234	79.62	0.34338	0.4878	-57.85	-37.47	-37.62	0.14	1.22	-0.36	0.65	-1.53	0.015875	OTHER	0.0000209	PROTEIN	234	2.63	177	313	42	3	0.59	0.431
hs_cds396	294	80.5	0.32141	0.51559	-130.97	-85.78	-85.63	-0.15	1.27	-0.83	0.65	-1.02	0.059955	OTHER	2.257E-08	PROTEIN	294	2.73	376	578	84	5	0.59	0.444
hs_cds397	157	84.71	0.26692	0.62348	-174.7	-122.53	-125.62	3.09	1.12	0.88	0.7	-3.22	0	OTHER	0.00008376	PROTEIN	157	2.53	404	739	109	7	0.59	0.401
hs_cds398	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs_cds399	104	75.71	0.44618	0.48586	-165.49	-62.53	-61.09	-1.45	1.5	-0.28	0.38	-2.68	0.000002	OTHER	0	PROTEIN	104	1.82	540	563	90	8	0.59	0.464
hs_cds400	236	80.93	0.34395	0.4726	-41.37	-21.1	-20.47	-0.63	1.43	0.33	0.51	-3.21	0	OTHER	2.676E-14	PROTEIN	236	2.09	178	207	36	1	0.59	0.449
hs_cds401	318	78.1	0.35296	0.47069	-57.22	-34.33	-36	1.67	1.12	0.06	0.6	-2.32	0.000012	OTHER	6.916E-08	PROTEIN	318	2.03	202	223	41	4	0.59	0.441
hs_cds402	2	88	0.11997	0.53247	-484.27	-432.72	-432.97	0.25	1.06	-1.5	0.89	0.15	0.5922	RNA	-	-	2	0.68	1240	1265	416	21	0.59	0.27
hs_cds403	164	87.57	0.23668	0.45274	-89.55	-64.93	-64.61	-0.32	1.25	-0.21	0.73	-1.8	0.007883	OTHER	3.554E-07	PROTEIN	164	1.61	313	338	76	6	0.59	0.404
hs_cds404	147	81.74	0.32352	0.42757	-272.93	-161.45	-164.62	3.17	1.26	0.01	0.59	-2.45	0.000006	OTHER	0	PROTEIN	147	1.67	898	1345	196	17	0.59	0.422
hs_cds405	54	78.04	0.39949	0.52933	-177.77	-86.14	-88.13	1.99	1.37	-0.3	0.48	-2.31	0.000013	OTHER	1.029E-12	PROTEIN	54	2.14	518	605	132	8	0.59	0.403
hs_cds406	2	95.04	0.04958	0.53837	-1235.31	-1193.13	-1193.38	0.25	1	1.11	0.97	-3.03	0	OTHER	-	-	2	0.66	3561	3569	1087	66	0.588	0.307
hs_cds407	3	98.49	0.02081	0.54636	-268.67	-260.17	-260.4	0.23	1.01	-0.02	0.97	-1.8	0.007972	OTHER	0.02	PROTEIN	3	0.61	706	730	232	14	0.59	0.274
hs_cds408	3	98.98	0.01022	0.43462	-206.66	-200.66	-202.91	2.25	1.01	-0.45														

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ec_cds432	33	79.27	0.37147	0.48	-22.07	-16.94	-15.78	-1.16	1.44	-4.2	0.77	4.12	0.999983	RNA	0.532	OTHER	33	3.15	55	72	17	0	1.011	0.837
ec_cds433	311	76.52	0.43771	0.5751	-513.26	-261.36	-258.82	-2.55	1.46	-1.5	0.51	-0.38	0.258292	OTHER	0	PROTEIN	311	4.21	1145	1211	279	22	0.59	0.466
ec_cds434	8	92.51	0.13426	0.47662	-303.66	-258.74	-260.34	1.6	1.16	-0.21	0.85	-1.64	0.011909	OTHER	0	PROTEIN	8	0.78	907	935	287	19	0.59	0.298
ec_cds435	70	88.91	0.19957	0.53764	-184.76	-130.97	-128.48	-2.49	1.21	0.28	0.71	-2.72	0.000001	OTHER	0.000000004	PROTEIN	70	1.11	498	522	161	7	0.59	0.315
ec_cds436	65	84.9	0.27786	0.5383	-253.32	-166.41	-169.41	2.99	1.2	0.77	0.66	-3.25	0	OTHER	6.22E-10	PROTEIN	65	0.99	698	734	223	13	0.589	0.306
ec_cds437	38	79.47	0.39331	0.51879	-83.83	-39	-41.54	2.54	1.35	0.02	0.47	-2.81	0.000001	OTHER	1.727E-08	PROTEIN	38	0.91	277	308	86	4	0.589	0.312
ec_cds438	330	85.51	0.26111	0.4896	-86.13	-55.24	-57.35	2.11	1.11	-2.13	0.64	0.35	0.70857	RNA	3.004E-07	PROTEIN	330	4.12	265	284	51	5	0.59	0.484
ec_cds439	94	79.27	0.38832	0.45795	-63.29	-29.49	-30.3	0.81	1.31	-0.1	0.47	-2.67	0.000002	OTHER	0.893	OTHER	94	2.07	219	240	51	3	0.59	0.421
ec_cds440	3	96.92	0.03075	0.53157	-378.08	-367.4	-366.65	-0.75	1.04	-1.64	0.97	0.27	0.662904	RNA	-	-	3	0.62	943	968	312	22	0.59	0.274
ec_cds441	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec_cds442	245	87.88	0.21969	0.56423	-199.94	-161.96	-163.07	1.1	1.24	-2.77	0.81	1.87	0.992958	RNA	2.665E-15	PROTEIN	245	3.07	468	489	146	9	0.59	0.398
ec_cds443	244	78.2	0.40657	0.57735	-192.55	-98.75	-98.5	-0.25	1.38	-2.58	0.51	0.88	0.910046	RNA	0	PROTEIN	244	3.73	445	470	121	7	0.59	0.435
ec_cds444	1140	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1140	5.19	917	1018	169	17	0.59	0.506
ec_cds445	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec_cds446	381	80.52	0.36206	0.60362	-384.65	-231.35	-226.75	-4.6	1.38	-1.85	0.6	0.24	0.647011	RNA	0	PROTEIN	381	5.57	818	861	166	20	0.59	0.493
ec_cds447	157	83.13	0.31995	0.59817	-44.62	-27.91	-29.03	1.11	1.31	-1.23	0.63	-0.61	0.160158	OTHER	0.00001153	PROTEIN	157	2.82	113	130	32	2	0.59	0.414
ec_cds448	12	97.6	0.04509	0.53182	-426.64	-405.65	-405.41	-0.24	1.06	-1.71	0.95	0.34	0.702764	RNA	4.331E-11	PROTEIN	12	0.62	1150	1179	381	26	0.59	0.272
ec_cds449	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec_cds450	46	74.76	0.4839	0.46552	-124.55	-40.35	-40.44	0.09	1.39	-0.78	0.32	-2.21	0.00269	OTHER	0	PROTEIN	46	1.42	397	427	96	10	0.59	0.396
ec_cds451	110	69.48	0.57573	0.64549	-284.35	-90.4	-90.71	0.31	1.56	-0.4	0.32	-2.29	0.000014	OTHER	0	PROTEIN	110	4.22	589	615	159	14	0.59	0.444
ec_cds452	88	78.31	0.41696	0.57868	-337.77	-179.95	-183.52	3.57	1.4	-1.58	0.53	-0.26	0.327476	OTHER	0	PROTEIN	88	1.39	787	811	224	14	0.59	0.368
ec_cds453	11	98.22	0.03303	0.50917	-592.3	-567.07	-569.14	2.07	1.05	-3.71	0.96	2.71	0.999252	RNA	2.094E-07	PROTEIN	11	0.6	1555	1583	522	31	0.589	0.267
ec_cds454	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec_cds455	116	81.65	0.34362	0.55923	-469.3	-274.91	-266.84	-8.08	1.45	-0.7	0.59	-1.42	0.021229	OTHER	0	PROTEIN	116	1.92	1144	1170	363	23	0.59	0.36
ec_cds456	63	81.92	0.33901	0.54545	-588.42	-334.97	-335.9	0.93	1.35	-0.95	0.57	-1.23	0.035212	OTHER	0	PROTEIN	63	1.1	1516	1545	503	26	0.59	0.311
ec_cds457	63	85.11	0.27054	0.53513	-296.6	-185.23	-183.75	-1.48	1.27	-1.55	0.62	-0.48	0.211143	OTHER	0	PROTEIN	63	2.48	769	802	215	13	0.59	0.406
ec_cds458	2	97.03	0.02969	0.56393	-385.8	-368.02	-367.52	-0.5	1.03	-1.23	0.95	-0.34	0.278733	OTHER	-	-	2	0.6	943	973	311	16	0.591	0.275
ec_cds459	22	80.71	0.36172	0.57241	-556.92	-300.63	-296.53	-4.1	1.39	-1.19	0.54	-0.97	0.067014	OTHER	0	PROTEIN	22	1.44	1288	1325	404	16	0.59	0.349
ec_cds460	68	71.01	0.54065	0.53351	-102.12	-35.25	-33.48	-1.76	1.45	-0.68	0.35	-1.92	0.005782	OTHER	0	PROTEIN	68	3.1	285	311	61	5	0.59	0.464
ec_cds461	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec_cds462	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds463	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds464	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds465	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds466	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds467	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds468	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds469	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds470	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds471	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds472	3	86.43	0.1357	0.32659	-158.95	-130.35	-130.6	0.25	1.12	-2.36	0.82	0.97	0.928403	RNA	-	-	3	0.81	619	648	199	11	0.589	0.288
ss_cds473	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds474	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds475	4	89.61	0.1039	0.2835	-79.11	-63.4	-66.4	3	1.11	-0.09	0.8	-2.21	0.002656	OTHER	-	-	4	0.83	385	407	118	4	0.59	0.299
ss_cds476	32	76.07	0.45528	0.39904	-243.71	-117.16	-116.23	-0.92	1.45	0.28	0.48	-2.8	0.000001	OTHER	0	PROTEIN	32	2.92	843	922	160	12	0.59	0.48
ss_cds477	4	67.15	0.5502	0.40708	-384.65	-135.96	-146.84	10.87	1.47	0.61	0.35	-3.5	0	OTHER	0	PROTEIN	4	1.47	1398	1423	418	24	0.59	0.359
ss_cds478	3	87	0.13004	0.31238	-47.1	-34.1	-34.1	0	1.11	-1.02	0.72	-1.31	0.0282	OTHER	-	-	3	0.84	223	243	63	5	0.591	0.32
ss_cds479	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds480	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds481	12	71.39	0.53702	0.36338	-130.85	-37.37	-36.22	-1.15	1.53	-0.46	0.29	-2.55	0.000004	OTHER	0	PROTEIN	12	2.08	538	560	126	9	0.59	0.428
ss_cds482	3	81.63	0.18372	0.31709	-192	-146.5	-145.75	-0.75	1.17	-0.72	0.76	-1.23	0.035354	OTHER	-	-	3	0.86	811	835	245	16	0.589	0.309
ss_cds483	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds484	3	83.26	0.16742	0.33069	-89.6	-71.1	-70.6	-0.5	1.14	1.09	0.79	-3.45	0	OTHER	-	-	3	0.86	442	465	132	8	0.59	0.309
ss_cds485	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds486	8	67.5	0.57593	0.362	-168.26	-47.92	-44.64	-3.28	1.69	0.57	0.28	-3.7	0	OTHER	0	PROTEIN	8	1.68	648	672	192	6	0.589	0.363
ss_cds487	2	85.46	0.1454	0.33468	-142.76	-111.61	-114.61	3	1.12	0.27	0.78	-2.58	0.000003	OTHER	-	-	2	0.74	619	643	219	10	0.589	0.256
ss_cds488	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds489	5	67.93	0.53054	0.3606	-265.66	-93.69	-104	10.31	1.45	0.8	0.35	-3.81	0	OTHER	0	PROTEIN	5	1.51	1057	1081	278	22	0.59	0.38
ss_cds490	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds491	4	71.55	0.39641	0.34918	-167.34	-72.5	-70.59	-1.91	1.35	-0.14	0.43	-2.81	0.000001	OTHER	0	PROTEIN	4	1.15	718	743	197	11	0.591	0.353
ss_cds492	2	86.03	0.13965	0.30879	-178.73	-146.47	-144.22	-2.25	1.17	0.21	0.82	-2.31	0.000012	OTHER	-	-	2	0.79	802	826	271	11	0.589	0.272
ss_cds493	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ss_cds494	6	68.6	0.56478	0.36569	-166.77	-40.2	-42.32	2.12	1.49	-0.78	0.24	-2.3	0.000013	OTHER	0	PROTEIN	6	1.62	662	685	174	8	0.59	0.388
ss_cds495	0	-	-	-	-																			

Table 3 – continued from previous page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
hs.5putr_520	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_521	13	91.02	0.16051	0.44516	-106.82	-83.47	-82.22	-1.24	1.12	-0.13	0.78	-2	0.004676	OTHER	0.929	OTHER	13	0.84	385	407	113	5	0.589	0.315	
hs.5putr_522	15	76.01	0.42775	0.47033	-16.11	-7.53	-7.62	0.09	1.44	0.32	0.47	-3.03	0	OTHER	0.974	OTHER	15	1.25	90	111	18	2	0.634	0.466	
hs.5putr_523	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_524	23	84.38	0.2728	0.70562	-75.93	-48.44	-47.3	-1.13	1.23	0.85	0.64	-3.48	0	OTHER	0.09	OTHER	23	0.82	163	183	52	2	0.591	0.304	
hs.5putr_525	15	79.72	0.36867	0.82239	-38.47	-31.29	-31.58	0.29	1.09	0.81	0.81	-2.04	0.004157	OTHER	0.872	OTHER	15	0.98	81	112	29	1	0.7	0.396	
hs.5putr_526	13	76.8	0.40568	0.82488	-74.23	-48.32	-49.27	0.95	1.14	0.16	0.65	-1.92	0.005743	OTHER	0.595	OTHER	13	0.77	113	208	37	1	0.589	0.299	
hs.5putr_527	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_528	11	84.3	0.28083	0.81588	-21.93	-18.55	-19.22	0.67	1.17	-0.04	0.85	-1.15	0.043045	OTHER	0.343	OTHER	11	2.47	52	67	18	0	1.066	0.859	
hs.5putr_529	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_530	21	85.6	0.26157	0.6696	-37.55	-25.22	-24.83	-0.39	1.21	0.46	0.67	-2.88	0.000001	OTHER	0.136	OTHER	21	0.85	95	112	31	2	0.602	0.314	
hs.5putr_531	16	83.38	0.29886	0.74948	-48.88	-33.34	-34.17	0.84	1.21	0.13	0.68	-2.22	0.002575	OTHER	0.74	OTHER	16	0.83	97	118	31	1	0.59	0.307	
hs.5putr_532	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_533	6	85.55	0.25722	0.47941	-100.35	-68.03	-69.15	1.13	1.07	-0.09	0.68	-2.13	0.003247	OTHER	0.384	OTHER	6	0.78	341	363	106	7	0.59	0.299	
hs.5putr_534	16	80.95	0.34034	0.71445	-27.13	-17.25	-18.28	1.03	1.22	0.58	0.64	-2.83	0.000001	OTHER	0.78	OTHER	16	1.85	66	81	21	0	0.851	0.624	
hs.5putr_535	16	79.55	0.35714	0.45642	-11.35	-3.46	-3.47	0	1.4	-0.01	0.31	-3.78	0	OTHER	0.351	OTHER	16	1.68	71	90	17	1	0.794	0.61	
hs.5putr_536	19	80.15	0.38442	0.48119	-49.67	-30.04	-32.77	2.72	1.09	-0.66	0.6	-1.23	0.035487	OTHER	0.353	OTHER	19	1.07	184	204	48	2	0.59	0.351	
hs.5putr_537	35	77.21	0.42103	0.67244	-65.08	-33.68	-34.16	0.48	1.47	-0.66	0.52	-1.51	0.016917	OTHER	0.791	OTHER	35	0.96	142	162	49	2	0.59	0.295	
hs.5putr_538	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_539	10	87.46	0.21393	0.72664	-21.78	-18.81	-18.42	-0.39	1.21	-0.84	0.86	-0.38	0.258255	OTHER	0.628	OTHER	10	3.09	47	62	15	0	1.173	1.01	
hs.5putr_540	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_541	9	70.96	0.48104	0.51728	-116.28	-58.9	-61.32	2.42	1.13	0.33	0.51	-2.58	0.000003	OTHER	0.25	OTHER	9	1.3	392	488	78	6	0.59	0.416	
hs.5putr_542	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_543	33	77.22	0.41875	0.70111	-46.5	-22.95	-23.24	0.28	1.39	0.17	0.49	-2.77	0.000001	OTHER	0.667	OTHER	33	1.04	103	136	32	2	0.59	0.333	
hs.5putr_544	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_545	7	89.17	0.19252	0.73237	-14	-10.49	-10.27	-0.22	1.2	0.63	0.75	-2.98	0	OTHER	0.215	OTHER	7	4.35	40	54	12	1	1.367	1.229	
hs.5putr_546	6	86.85	0.21652	0.74853	-27.96	-20.8	-21.64	0.84	1.06	0.39	0.74	-2.62	0.000002	OTHER	0.356	OTHER	6	1.77	65	81	20	1	0.863	0.647	
hs.5putr_547	9	93.3	0.10644	0.77155	-206.78	-185.36	-185.55	0.19	1.09	0.06	0.9	-1.86	0.006811	OTHER	0.117	OTHER	9	0.52	342	373	122	7	0.591	0.246	
hs.5putr_548	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_549	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_550	7	94.95	0.09209	0.48773	-63.33	-55.49	-58.1	2.61	1.02	-1.98	0.88	0.57	0.815994	RNA	0.164	OTHER	7	0.77	185	204	54	5	0.589	0.315	
hs.5putr_551	15	80.13	0.34941	0.71912	-32.5	-23.22	-24	0.78	1.24	0.49	0.71	-2.28	0.000015	OTHER	0.898	OTHER	15	1.6	73	89	21	0	0.772	0.557	
hs.5putr_552	10	94.34	0.09986	0.80338	-26.13	-26.03	-25.78	-0.25	1.15	1.04	1	-2.57	0.000003	OTHER	0.677	OTHER	10	1.59	59	75	19	1	0.946	0.714	
hs.5putr_553	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_554	10	93.55	0.1103	0.8257	-20.13	-17.05	-17.22	0.17	1	1.19	0.85	-3.5	0	OTHER	0.369	OTHER	10	2.29	49	64	15	1	1.127	0.937	
hs.5putr_555	18	77.09	0.41037	0.4824	-16.04	-6.22	-5.25	-0.97	1.54	0.51	0.39	-3.78	0	OTHER	0.988	OTHER	18	1.38	86	104	14	1	0.662	0.531	
hs.5putr_556	49	85.85	0.22758	0.46889	-202.35	-171.94	-173.75	1.81	1.04	0.15	0.85	-1.65	0.011565	OTHER	0.031	PROTEIN	49	2.28	617	1108	150	10	0.59	0.405	
hs.5putr_557	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_558	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_559	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_560	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.5putr_561	7	94.4	0.10179	0.50263	-52.58	-45.21	-47.65	2.44	1.02	-1.45	0.86	-0.16	0.387846	OTHER	0.204	OTHER	7	0.78	161	180	46	4	0.59	0.325	
hs.5putr_562	10	88.22	0.2011	0.72299	-23.17	-20.54	-19.9	-0.64	1.25	-0.81	0.89	-0.33	0.287113	OTHER	0.528	OTHER	10	2.65	50	65	16	0	1.106	0.926	
hs.3putr_563	10	92.23	0.13973	0.41497	-60.24	-47.58	-49.45	1.86	1.06	0.43	0.79	-2.75	0.000001	OTHER	0.077	OTHER	10	0.92	259	281	66	4	0.588	0.358	
hs.3putr_564	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_565	36	84.45	0.28218	0.40395	-263.04	-179.95	-180.16	0.21	1.15	-1.11	0.68	-0.67	0.139103	OTHER	0.164	OTHER	36	1.29	954	1149	294	19	0.591	0.326	
hs.3putr_566	103	78.74	0.38257	0.40738	-40.88	-22.77	-23.13	0.36	1.12	0.56	0.56	-3.05	0	OTHER	0.329	OTHER	103	2.57	194	303	33	2	0.59	0.474	
hs.3putr_567	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_568	28	81.04	0.30548	0.47	-120.99	-92.53	-93.32	0.79	1.18	-0.86	0.76	-0.44	0.228909	OTHER	0.352	OTHER	28	1.02	362	530	105	8	0.59	0.378	
hs.3putr_569	26	86.03	0.25251	0.45694	-158.51	-91.91	-90.4	-1.51	1.22	0.09	0.58	-2.94	0	OTHER	0.433	OTHER	26	2.08	523	773	159	8	0.59	0.356	
hs.3putr_570	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_571	11	81.05	0.36083	0.48244	-547.63	-357.85	-362.48	4.63	1.22	-0.78	0.65	-0.89	0.081532	OTHER	0.888	OTHER	11	0.81	1602	1628	523	29	0.59	0.289	
hs.3putr_572	12	91.04	0.15956	0.42004	-61.52	-43.24	-44.41	1.17	1.07	0.33	0.7	-3.02	0	OTHER	0.049	PROTEIN	12	0.87	258	280	72	3	0.59	0.333	
hs.3putr_573	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_574	14	81.86	0.32852	0.46058	-22.43	-8.34	-8.67	0.33	1.2	-0.25	0.37	-3.3	0	OTHER	0.686	OTHER	14	1.12	106	124	26	1	0.591	0.371	
hs.3putr_575	76	76.73	0.39135	0.27577	-115.72	-42.92	-40.16	-2.76	1.32	0.72	0.37	-4.23	0	OTHER	0.074	OTHER	76	2.5	609	679	121	8	0.59	0.44	
hs.3putr_576	86	85.62	0.2492	0.28484	-21.3	-13.73	-13.82	0.08	1.1	-0.48	0.64	-1.88	0.0063	OTHER	0.937	OTHER	86	2	125	151	32	1	0.59	0.392	
hs.3putr_577	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_578	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hs.3putr_579	12	91.04	0.15956	0.42004	-61.52	-43.24	-44.41	1.17	1.07	0.33	0.7	-3.02	0	OTHER	0.09	OTHER	12	0.87	258	280	72	3	0.59	0.333	
hs.3putr_580	0	-	-	-	-																				

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
hs-3putr-608	77	82.44	0.32921	0.23088	-19.24	-8.18	-7.55	-0.63	1.38	0.06	0.43	-3.37	0	OTHER	0.478	OTHER	77	2.14	151	182	31	2	0.59	0.431	
hs-3putr-609	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.59	171	191	55	1	0.589	0.277	
hs-3putr-610	32	85.36	0.26017	0.43457	-82.7	-43.51	-43.27	-0.24	1.22	-0.55	0.53	-2.35	0.00001	OTHER	0.349	OTHER	32	2.19	310	353	78	5	0.59	0.401	
hs-3putr-611	107	79.81	0.36456	0.41048	-48.09	-27.39	-27.15	-0.24	1.26	0.26	0.57	-2.69	0.000002	OTHER	0.413	OTHER	107	2.32	218	311	39	4	0.59	0.461	
hs-3putr-612	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.75	298	319	85	3	0.59	0.322	
ec-5putr-613	8	90.8	0.14638	0.42611	-4.9	-4.2	-4.83	0.63	1.11	-0.6	0.86	-1.03	0.05832	OTHER	0.342	OTHER	8	29	41	9	0	1.59	1.518		
ec-5putr-614	17	83.66	0.28902	0.4513	-45.55	-37.42	-36.35	-1.07	1.31	-1.78	0.82	1.01	0.934612	RNA	0.31	OTHER	17	0.98	148	166	38	1	0.589	0.363	
ec-5putr-615	26	79.11	0.40182	0.31789	-14.37	-7.29	-7.65	0.36	1.29	-0.58	0.51	-1.76	0.008719	OTHER	0.578	OTHER	26	1.77	84	100	20	0	0.676	0.483	
ec-5putr-616	14	87.31	0.21928	0.29231	-16.16	-12.68	-12.23	-0.44	1.12	-1.12	0.78	-0.42	0.240778	OTHER	0.069	OTHER	14	0.82	108	142	29	1	0.591	0.335	
ec-5putr-617	31	86.79	0.2297	0.35517	-4.47	-4.14	-3.78	-0.36	1.25	-0.78	0.93	-0.02	0.480396	OTHER	0.66	OTHER	31	11.87	34	47	9	0	1.595	1.542	
ec-5putr-618	2	81.82	0.18182	0.4482	-2.9	-1.35	-1.35	0	1	0.92	0.47	-4.87	0	OTHER	-	-	2	2	44	60	7	1	0.929	0.811	
ec-5putr-619	110	78.96	0.40428	0.35141	-25.15	-15.13	-16.05	0.92	1.28	-1.94	0.6	0.56	0.812661	RNA	0.998	OTHER	110	1.61	102	132	27	2	0.59	0.381	
ec-5putr-620	47	81.24	0.36057	0.37245	-15.32	-9.68	-10.15	0.47	1.23	-0.32	0.63	-1.62	0.012752	OTHER	0.381	OTHER	47	2.52	83	99	15	0	0.684	0.562	
ec-5putr-621	10	93.94	0.09834	0.32292	-0.79	0	0	0	0	0.78	0	-6.9	0	OTHER	0.337	OTHER	10	6.74	33	47	6	0	1.641	1.592	
ec-5putr-622	57	80.45	0.37093	0.42294	-21.53	-16	-15.53	-0.47	1.53	-0.74	0.74	-0.39	0.253651	OTHER	0.308	OTHER	57	1.97	89	106	17	0	0.64	0.501	
ec-5putr-623	72	77.68	0.42447	0.425	-25.42	-17.27	-17.47	0.2	1.32	-0.57	0.68	-0.7	0.130656	OTHER	0.723	OTHER	72	1.24	117	135	25	0	0.589	0.405	
ec-5putr-624	25	86.9	0.23966	0.3902	-4.35	-3.57	-3.57	0	1	-0.26	0.82	-1.23	0.035291	OTHER	0.863	OTHER	25	4.23	42	56	4	0	1.305	1.266	
ec-5putr-625	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ec-5putr-626	24	85.21	0.26689	0.40124	-11.92	-6.57	-6.98	0.42	1.17	0.9	0.55	-4.06	0	OTHER	0.792	OTHER	24	1.41	80	95	26	0	0.708	0.434	
ec-5putr-627	4	96.15	0.03846	0.25462	-2.1	-2.1	-2.1	0	1	-0.93	1	-0.43	0.235121	OTHER	-	-	4	4	26	40	3	0	1.426	1.385	
ec-5putr-628	62	74.28	0.45444	0.29473	-88.68	-42.32	-40.92	-1.4	1.58	-1.22	0.48	-0.85	0.091017	OTHER	4.063E-14	PROTEIN	62	1.57	293	500	70	7	0.59	0.399	
ec-5putr-629	32	76.98	0.4352	0.44981	-19.53	-11.74	-11.69	-0.05	1.35	-0.33	0.6	-1.42	0.021381	OTHER	0.15	OTHER	32	1.61	96	116	18	2	0.595	0.437	
ec-5putr-630	60	79.96	0.37524	0.61403	-71.2	-36.84	-37.43	0.59	1.37	0.3	0.52	-2.97	0	OTHER	7.772E-16	PROTEIN	60	1.34	188	207	47	2	0.591	0.393	
ec-5putr-631	60	80.07	0.37325	0.61078	-71.42	-36.84	-37.43	0.59	1.37	0.26	0.52	-2.93	0	OTHER	1.118E-13	PROTEIN	60	1.34	189	208	48	2	0.59	0.388	
ec-5putr-632	22	92.52	0.14395	0.37425	-54.54	-45.84	-45.04	-0.8	1.2	-2.42	0.84	1.19	0.957459	RNA	0.837	OTHER	22	0.81	212	232	65	4	0.589	0.301	
ec-5putr-633	24	92.22	0.15037	0.38806	-45.56	-40.1	-39.91	-0.19	1.18	-3.33	0.88	2.54	0.998812	RNA	0.997	OTHER	24	0.88	179	199	48	2	0.591	0.34	
ec-5putr-634	6	89.06	0.20076	0.45302	-2.36	-2	-2	0	1	0.79	0.85	-2.6	0.000003	OTHER	0.656	OTHER	6	6	32	46	5	0	1.44	1.377	
ec-5putr-635	19	92.54	0.14579	0.4364	-28.88	-25	-25.25	0.25	1.12	-2.57	0.87	1.54	0.983026	RNA	0.926	OTHER	19	0.78	92	110	30	1	0.621	0.317	
ec-5putr-636	27	87.8	0.23299	0.3675	-73.86	-52.41	-53.59	1.18	1.24	-0.93	0.71	-0.98	0.06556	OTHER	0.316	OTHER	27	0.99	317	339	84	5	0.59	0.35	
ec-5putr-637	10	92.41	0.13306	0.36798	-2.34	-1.62	-1.22	-0.4	1.25	0.18	0.69	-3.01	0	OTHER	0.486	OTHER	10	10	29	42	8	0	1.689	1.636	
ec-5putr-638	36	74.72	0.47348	0.34504	-53	-17.33	-16.7	-0.63	1.49	-1.94	0.33	-0.7	0.128393	OTHER	0.873	OTHER	36	1.36	216	235	59	2	0.59	0.358	
ec-5putr-639	16	86.65	0.25482	0.33821	-44.69	-28.44	-26.78	-1.66	1.33	-1.97	0.64	0.11	0.561664	RNA	0.104	OTHER	16	0.94	182	213	60	1	0.59	0.288	
ec-5putr-640	24	84.03	0.30045	0.39891	-12.47	-6.79	-6.07	-0.72	1.31	-0.32	0.54	-2.41	0.000007	OTHER	0.855	OTHER	24	2.44	66	82	17	0	0.85	0.685	
ec-5putr-641	49	80.42	0.31033	0.50952	-28.92	-15.54	-14.9	-0.64	1.17	-0.22	0.54	-2.49	0.000005	OTHER	0.647	OTHER	49	2.38	106	123	25	1	0.59	0.422	
ec-5putr-642	297	75.77	0.39456	0.5435	-136.65	-79.97	-80.96	0.99	1.29	-1.06	0.59	-0.7	0.128695	OTHER	2.217E-10	PROTEIN	297	3.13	358	378	92	7	0.59	0.433	
ec-5putr-643	50	83.39	0.29579	0.46154	-62.74	-34.45	-34.51	0.06	1.24	-1	0.55	-1.48	0.018273	OTHER	0.091	OTHER	50	1.62	220	267	48	2	0.59	0.419	
ec-5putr-644	189	80.12	0.37893	0.53397	-56.38	-23.75	-25.95	2.2	1.23	-1.23	0.42	-1.53	0.016195	OTHER	1.008E-13	PROTEIN	189	2.43	159	177	42	3	0.59	0.424	
ec-5putr-645	11	85.33	0.26568	0.44012	-4.53	-1.21	-0.68	-0.53	1.75	1.63	0.27	-6.27	0	OTHER	0.302	OTHER	11	3.09	49	63	11	0	1.128	1.005	
ec-5putr-646	30	83.9	0.30365	0.49062	-1.84	0	0	0	0	0.75	0	-6.21	0	OTHER	0.745	OTHER	30	8.89	39	53	4	0	1.4	1.379	
ec-5putr-647	183	85.71	0.26716	0.54021	-150.88	-106.17	-104.8	-1.37	1.33	-0.8	0.7	-1.04	0.056479	OTHER	5.945E-13	PROTEIN	183	2.76	415	490	108	6	0.59	0.426	
ec-5putr-648	62	83.57	0.32608	0.50481	-24.35	-14.37	-14.77	0.39	1.24	-0.93	0.75	-0.59	-1.44	0.002075	OTHER	0.065	OTHER	62	1.72	86	104	22	0	0.662	0.465
ec-5putr-649	200	81.1	0.36185	0.54199	-11.37	-61.9	-62.63	0.73	1.29	-0.77	0.56	-1.41	0.021718	OTHER	4.774E-15	PROTEIN	200	2.68	288	308	72	5	0.59	0.426	
ec-5putr-650	4	93.94	0.08348	0.47691	-1.03	-0.9	-1.23	0.33	1	-0.17	0.67	-2.9	0.000001	OTHER	0.306	OTHER	4	4	22	36	5	0	1.461	1.361	
ec-5putr-651	250	77.15	0.40556	0.51936	-139.82	-76.79	-77.12	0.33	1.35	-1.46	0.55	-0.35	0.273931	OTHER	0	PROTEIN	250	4.82	404	494	95	7	0.59	0.47	
ec-5putr-652	250	79.81	0.38331	0.60376	-151.04	-93.74	-92.2	-1.54	1.39	-1.03	0.62	-0.63	0.154103	OTHER	0	PROTEIN	250	2.58	343	369	83	9	0.59	0.436	
ec-5putr-653	254	77.57	0.41643	0.53313	-251.91	-143.74	-145.68	1.94	1.38	-1.36	0.57	-0.32	0.292691	OTHER	0	PROTEIN	254	2.11	648	782	179	14	0.59	0.397	
ec-5putr-654	6	93.94	0.09834	0.30019	-2.08	0	0	0	0	-0.71	0	-5.3	0	OTHER	0.445	OTHER	6	6	33	47	6	0	1.612	1.542	
ec-5putr-655	26	85.24	0.2895	0.38316	-34.47	-15.37	-16.41	1.03	1.27	-0.71	0.45	-2.45	0.000006	OTHER	0.993	OTHER	26	1.08	174	193	40	1	0.59	0.387	
ec-5putr-656	2	96	0.04	0.3875	-0.65	-0.5	-0.5	0	1	0.57	0.77	-3.49	0	OTHER	-	-	2	2	25	40	4	0	0.977	0.864	
ec-5putr-657	93	81.57	0.3507	0.57414	-78.98	-47.83	-47.08	-0.74	1.39	-1.6	0.61	-0.08	0.43613	OTHER	7.435E-09	PROTEIN	93	1.69	195	214	57	3	0.59	0.365	
ec-5putr-658	2	80.77	0.19231	0.37308	-1.45	-0.95	-0.95	0	1	0.06	0.66	-2.75	0.000001	OTHER	-	-	2	2	26	41	4	0	0.857	0.744	
ec-5putr-659	89	82.44	0.33156	0.51235	-125.93	-78.93	-78.05	-0.88	1.32	-0.98	0.63	-0.88	0.083724	OTHER	2.998E-15	PROTEIN	89	1.8	359	384	84	8	0.59	0.417	
ec-5putr-660	254	83.1	0.32268	0.54018	-213.25	-139.42	-139.72	0.3	1.33	-0.78	0.65	-1.08	0.051542	OTHER	0	PROTEIN	254	3.32	548	633	139	11	0.59	0.439	
ec-5putr-661	106	79.97	0.37735	0.48516	-48.67	-30.41	-30.38	-0.22	1.19	-1.42	0.62	-0.14	0.401024	OTHER	0.000009461	PROTEIN	106	1.87	165	187	29				

Table 3 – continued from previous page

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ec_3putr_696	5	91.77	0.13538	0.5147	-35.72	-35.7	-35.2	-0.5	1.16	-5.73	1	5.44	0.999999	RNA	0.992	OTHER	5	1.12	77	93	26	1	0.733	0.435
ec_3putr_697	70	83.08	0.33097	0.42827	-21.95	-18.78	-20.67	1.89	1.07	-5.52	0.86	5.69	1	RNA	0.083	OTHER	70	3.7	46	68	17	1	1.198	0.995
ec_3putr_698	209	86.65	0.22301	0.57768	-19.67	-16.6	-16.6	0	1	-4.67	0.84	4.24	0.999987	RNA	0.118	OTHER	209	209	28	85	11	0	1.53	1.225
ec_3putr_699	14	93.44	0.12758	0.45678	-25.38	-25.33	-25.22	-0.11	1.11	-4.97	1	4.7	0.999996	RNA	0.384	OTHER	14	1.7	60	76	19	1	0.931	0.678
ec_3putr_700	18	83.8	0.28603	0.4945	-27.3	-24.13	-25.43	1.31	1.1	-4.75	0.88	4.82	0.999997	RNA	0.831	OTHER	18	1.31	76	93	21	2	0.743	0.501
ec_3putr_701	13	91.58	0.15036	0.4783	-18.98	-19.12	-18.98	-0.14	1.15	-3.78	1.01	3.63	0.999935	RNA	0.311	OTHER	13	4.28	39	52	14	0	1.4	1.252
ec_3putr_702	8	97.22	0.04507	0.42794	-16.38	-16.75	-16.38	-0.38	1.14	-2.76	1.02	1.94	0.994105	RNA	0.375	OTHER	8	1.9	54	69	16	1	1.027	0.805
ec_3putr_703	24	79.63	0.3742	0.51288	-14.57	-10.68	-11.02	0.33	1	-2.6	0.73	2.01	0.995079	RNA	0.204	OTHER	24	4.72	46	77	11	1	1.197	1.09
ec_3putr_704	42	80.14	0.38538	0.56406	-23.02	-20.48	-20.37	-0.11	1.08	-4.79	0.89	5.37	0.999999	RNA	0.797	OTHER	42	8.04	47	62	12	0	1.173	1.062
ec_3putr_705	5	62.53	0.60602	0.54072	-111.92	-82.64	-81.45	-1.19	1.19	-6.08	0.74	6.75	1	RNA	0.268	OTHER	5	1.13	243	279	76	5	0.59	0.315
ec_3putr_706	21	89.47	0.18397	0.37186	-21.18	-20.71	-21.1	0.39	1.07	-7.12	0.98	6.66	1	RNA	0.858	OTHER	21	2.79	47	61	14	0	1.173	0.99
ec_3putr_707	10	92.97	0.1238	0.50556	-20.56	-20.14	-20.22	0.08	1.08	-6.22	0.98	5.71	1	RNA	0.76	OTHER	10	4.71	37	51	12	0	1.472	1.349
ec_3putr_708	24	83.68	0.3139	0.55579	-13.98	-13.98	-13.98	0	1	-2.67	1	3.14	0.999759	RNA	0.906	OTHER	24	24	32	44	7	0	1.617	1.591
ec_3putr_709	191	81.28	0.3428	0.49671	-59.19	-42.89	-42.95	0.06	1.16	-2.89	0.72	2.17	0.996782	RNA	9.228E-07	PROTEIN	191	2.06	165	193	45	2	0.59	0.392
ec_3putr_710	46	76.97	0.41504	0.45547	-22.67	-18.6	-18.55	-0.05	1.08	-4.94	0.82	5.35	0.999999	RNA	0.191	OTHER	46	2.78	57	75	16	0	0.977	0.796
ec_3putr_711	5	87.62	0.17647	0.4123	-17.63	-18.32	-18.1	-0.22	1.09	-4.58	1.04	4.72	0.999996	RNA	0.525	OTHER	5	5	35	48	11	0	1.456	1.344
ec_3putr_712	15	84.81	0.26892	0.70876	-20.07	-18.3	-18.3	0	1	-3.61	0.91	3.62	0.999933	RNA	0.809	OTHER	15	9.39	34	47	12	0	1.595	1.483

References

1. Statistics of of sequence similarity scores. <http://www.ncbi.nlm.nih.gov/BLAST/tutorial/>. Accessed: 2016-03-19.
2. David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 35(suppl 1):D5–D12, 2007.
3. Travis J Wheeler, Jody Clements, Sean R Eddy, Robert Hubley, Thomas A Jones, Jerzy Jurka, Arian FA Smit, and Robert D Finn. Dfam: a database of repetitive dna based on profile hidden markov models. *Nucleic acids research*, 41(D1):D70–D82, 2013.
4. Heladia Salgado, Martin Peralta-Gil, Socorro Gama-Castro, Alberto Santos-Zavaleta, Luis Muñoz-Rascado, Jair S García-Sotelo, Verena Weiss, Hilda Solano-Lira, Irma Martínez-Flores, Alejandra Medina-Rivera, et al. Regulondb v8. 0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic acids research*, 41(D1):D203–D213, 2013.
5. Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. Genbank. *Nucleic acids research*, page gks1195, 2012.
6. Andrew Yates, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Konstantinos Bilis, Denise Carvalho-Silva, Carla Cummins, Peter Clapham, Stephen Fitzgerald, Laurent Gil, et al. Ensembl 2016. *Nucleic acids research*, 44(D1):D710–D716, 2016.

7 Paper: CMCompare webserver: comparing RNA families via covariance models

Florian Eggenhofer, Ivo L. Hofacker, Christian Höner zu Siederdisen

CMCompare webserver: comparing *RNA* families via covariance models.

Nucl. Acids Res. (1 July 2013) 41 (W1): W499-W503.

Florian Eggenhofer wrote the software. All three authors participated in writing the paper.

Bibliography:

Florian Eggenhofer, Ivo L. Hofacker, and Christian Höner zu Siederdisen. CMCompare webserver: comparing RNA families via covariance models, Nucl. Acids Res. (1 July 2013) 41 (W1): W499-W503 first published online May 2, 2013 doi:10.1093/nar/gkt329

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

CMCompare webserver: comparing RNA families via covariance models

Florian Eggenhofer^{1,*}, Ivo L. Hofacker^{1,2} and Christian Höner zu Siederdissen^{1,*}

¹Institute for Theoretical Chemistry, University of Vienna, Währingerstrasse 17, A-1090 Vienna, Austria and

²Bioinformatics and Computational Biology Research Group, University of Vienna, Währingerstrasse 17, A-1090 Vienna, Austria

Received January 31, 2013; Revised April 2, 2013; Accepted April 9, 2013

ABSTRACT

A standard method for the identification of novel non-coding RNAs is homology search by covariance models. Covariance models are constructed for specific RNA families with common sequence and structure (e.g. transfer RNAs). Currently, there are models for 2208 families available from Rfam. Before being included into a database, a proposed family should be tested for specificity (finding only true homolog sequences), sensitivity (finding remote homologs) and uniqueness. The CMCompare webserver (CMCws) compares Infernal RNA family models to (i) identify models with poor specificity and (ii) explore the relationship between models. The CMCws provides options to compare new models against all existing models in the current Rfam database to avoid the construction of duplicate models for the same non-coding RNA family. In addition, the user can explore the relationship between two or more models, including whole sets of user-created family models. Visualization of family relationships provides help in evaluating candidates for clusters of biologically related families, called clans. The CMCws is freely available, without any login requirements, at <http://rna.tbi.univie.ac.at/cmcws>, and the underlying software is available under the GPL-3 license.

INTRODUCTION

In the past years, and especially with the development of high-throughput methods like RNA sequencing, the scientific community became more and more aware of the importance of non-coding RNAs. These transcripts are found in all domains of life and regulate essential pathways and cellular processes.

Homologs of known RNA sequences can be detected in genomes using a number of methods. For close homologs, sequence-based methods like Blast (1) provide an extremely efficient search method. More remote homologs accumulate mutations on the sequence level, whereas the structure tends to be conserved. In structural non-coding RNAs, most of the statistical information appears to be available with the sequence and secondary structure. Methods like Infernal (2,3) can be used to transform the structural alignment of an RNA family of related sequences into a stochastic model called a covariance model.

RNA family models allow one to find new homolog family members by considering the structure and sequence features of this family. The number of covariance models, which is available from databases like Rfam (4,5), is constantly increasing.

Putative homologs discovered in a genome should, in principle, show strong affinity to only a single RNA family or, by extension, covariance model. In practice, some RNA families [e.g. RNaseP, rRNA (SSU)] have been intentionally split along kingdoms to preserve statistical signals owing to diverse sequence mutations and structural changes.

The CMCompare webserver (CMCws) provides an easy-to-use interface to check the discriminatory power of newly proposed RNA family models. This makes it possible to check that a similar model does not already exist in the database or that a set of existing or newly proposed models is not too closely related to each other in terms of the sequences they accept as putative homologs.

DESCRIPTION OF THE WEBSERVER

Functionality of CMCws

For newly constructed covariance models, it is useful to check what other models are already available in Rfam and compare them with each other. The CMCws is

*To whom correspondence should be addressed. Tel: +43 1 4277 52731; Fax: +43 1 4277 52793; Email: egg@tbi.univie.ac.at
Correspondence may also be addressed to Christian Höner zu Siederdissen. Tel: +43 1 4277 52737; Fax: +43 1 4277 52793;
Email: choener@tbi.univie.ac.at

based on 'CMCompare' (6), which returns a Link score for every pair of models checked. Link sequences and their associated Link scores are sequences giving high scores in both models simultaneously. A sequence with a Link score of, say 20 bits, scores at least 20 bits in each of the models. The Link sequence is the sequence with highest overall Link score (6). A high Link score can be an indicator for the following:

- (1) A model for the same RNA family is already present in the database. Using a curated model from Rfam avoids repetitive model construction and fine tuning. Also, improvements and extensions can be easier shared by finding and using a common set of models. Detection of a similar model by CMCws allows one to use this model instead.
- (2) At least one of the models lacks specificity, meaning that both score high for the same sequence. A model should detect only homologs belonging to the RNA family it represents, but not of member sequences of other families. During model construction, more members belonging to the RNA family are added to ensure detection over bigger phylogenetic distance, which can expand the space of detected sequences and associated structures to overlap with other families. By highlighting these overlaps, CMCws makes it possible to address this lack of specificity.
- (3) A biological relationship exists between the models that explain the overlap. Families derived from a common ancestor can share sequence and structure features. Rfam groups families related in this way as clans (7), which has been done up to now in a manual process. CMCws would allow Rfam to find possible candidates for clan members.

Input

After choosing the mode of comparison, the web server accepts a file upload containing one or more Infernal covariance models (Infernal 1.0 or later, Rfam 9 or later) or structural alignments using the Stockholm format as input. Stockholm alignments are internally converted to covariance models for further processing.

Processing

The web server relies on CMCompare (6), which is the first published tool for comparison of covariance models and has already been used in other projects (8,9). CMCompare has been expanded to also compare models created with Infernal 1.1 since publication.

Two modes of processing are available. The first mode allows one to compare the input models against all available models in Rfam or all models of specified subtype (micro RNAs, tRNAs) thereof, which reduces computation time. Alternatively, the set of uploaded models can be compared against each other.

Output

The first mode provides the user with a table of pairwise comparisons against Rfam models, as shown in Figure 1.

The result list, computed by CMCompare, can be filtered by model name, Link score and number of models. Each of the columns can be sorted. These filtering options allow one to easily extract similar models. A weighted graph representation visualizes selected models as nodes, and their Link scores as edges to simplify evaluation, see Figure 2b. By clicking on the edges or the magnifying glass icon, each pairwise comparison can also be viewed in detail, providing the common highest scoring sequence (Link sequence), corresponding structure and further information.

Models of interest from the result list, or a set of models that have been uploaded by the user, can be analyzed with the second mode. This mode returns all pairwise comparisons, which can also be sorted and filtered by the name of a second model. Exploring this list is especially useful to identify groups of models that are closely related and pose potential candidates for clans.

The output is visualized as a graph, as well as a matrix, which gives all pairwise Link scores, simplifying the identification of relevant links, see Figure 2. Comparing models cannot always capture the biological relationship between models, e.g. in the RNase P clan. Although the two different models for bacterial RNaseP are linked with each other, one of them is strongly linked with the corresponding RNaseP model for archae, and the other one is not. By using a graph representation, we are still indirectly able to identify potential clan members.

As noted before, Rfam clans are constructed entirely manually. We believe that CMCws can significantly facilitate this process.


Usage example

Assume we are interested in RNA families related with tRNAs. For this usage example, which follows Figures 1 and 2, we use as input the tRNA model (RF00005) from Rfam.

The first step is to select the comparison versus Rfam mode and upload the model to check for similar models already available from the database.

The top five resulting hits are shown in Figure 1, starting with tRNA having the maximal score possible with this model, compared with itself. The next models have Link scores between 10 and 20, indicating a moderate overlap between them.

For each of these models, one should investigate the reason for the high Link score, with potential reasons given previously as points 1–3. In decreasing order of Link score, we first consider the tRNA-Sec RNA family. Careful comparison of both secondary structures yields notable differences, including an additional stem in tRNA-Sec, but also some commonality. Based on commonalities and differences in biological action in the cell, as well as the differences in the structural alignment, one will probably not want to join both tRNA and tRNA-Sec, as a single family, but the commonalities are large enough to suggest a common Rfam clan, which for

cm vs cm Select covariance models from the list for comparison with each other 








		Rank	Link score	Rfam Id	Rfam Name	Input score	Rfam score
<input type="checkbox"/>		1.	83.555	RF00005	tRNA	83.555	83.555
<input type="checkbox"/>		2.	18.526	RF01852	tRNA-Sec	18.526	22.214
<input type="checkbox"/>		3.	13.064	RF01745	manA	16.322	13.064
<input type="checkbox"/>		4.	12.984	RF00023	tmRNA	12.984	16.842
<input type="checkbox"/>		5.	11.843	RF01850	beta_tmRNA	12.325	11.843

Figure 1. List of results: contains comparison results corresponding to the current filtering options. The list is sortable by all column names. The magnifying glass links to a detailed view of each comparison. The checkboxes on the right allow to select the models for a comparison with each other. CMCompare computes a score for the Input model (Input score) and for the Rfam model (Rfam score). The lower one is the Link score.

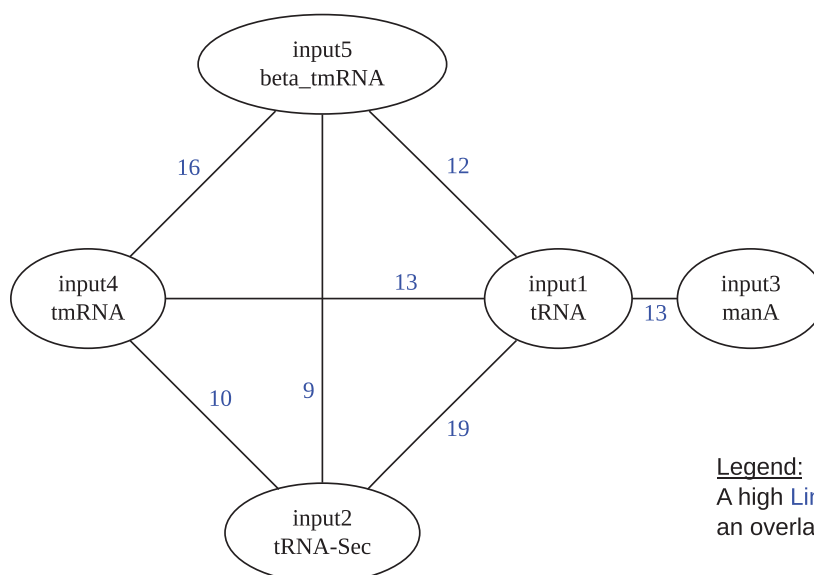
Matrix of result linkscores for all models: 

(a)

	Model 1	Model 2	Model 3	Model 4	Model 5
Model 1	x	18.526	13.064	12.984	11.843
Model 2	18.526	x	3.300	9.869	8.906
Model 3	13.064	3.300	x	-0.311	1.038
Model 4	12.984	9.869	-0.311	x	15.776
Model 5	11.843	8.906	1.038	15.776	x

▲ Hide

(b)



Legend:

A high Link score indicates an overlap in specificity

Figure 2. Visualizations: simplifying identification of relevant similarities between models by giving different representations of the pairwise result Link scores. (a) Link score matrix containing the similarity between all provided models and highlighting them by color. Clicking the Link score shows a detailed view of the comparison. (b) Weighted graph representation of linked models. The nodes indicate the models and contain their identifier. In contrast to the matrix representation, the shown edges correspond with the applied filtering options and redirect to a detailed view of the comparison on clicking. The comparisons against Rfam only show edges between the input and Rfam models. The shown input models 1, 2 and 4, 5 are members of the tRNA-clan, whereas ManA is presumably a false link.

Rfam is true. Incidentally, the CMCompare algorithm proposes a consensus secondary structure of both RNA families for the link sequence, which contains a total of three stems, with one tRNA and two tRNA-Sec stems deleted in the consensus.

The next two models in the list tmRNA and beta_tmRNA have a significantly lower Link score than the tRNA compared with itself but capture the similarity between the models. As an aside, both tmRNAs have a higher Link score between each other than to the tRNA model.

The final model flagged by the CMCws is the manA RNA motif family. The Link score is low (13 bits) so that no immediate action is warranted.

However, the nature of the manA and its secondary structure (the CMCompare algorithm proposes a low-scored cloverleaf consensus structure between the tRNA and manA families) makes it a candidate for further investigation. According to Rfam, this is a computationally identified RNA family that occurs often adjacent to tRNAs (10).

Among the first five hits of the list, we can find three of the five other members of the tRNA clan. To get a better idea about their relationship with each other, we can select and resubmit them to a cm versus cm comparison. Figure 2b shows the result for the submission of the top five models. The matrix representation gives an overview over all comparisons between the submitted models, whereas the weighted graph only shows RNA family models as nodes and linkscores as edges. As expected, we can see that there is a strong connection between the members of this clan and especially between the tmRNA models. The manA is only linked with the tRNA model, but not with the other clan members. The combination of these two comparison modes simplifies finding candidates for clan construction.

Following these conclusions, the tRNA family would be submitted for inclusion in the Rfam database, pointing out it is possible biological relationship with the tRNA-Sec family.

Implementation details

CMCws was implemented in Perl 5 using CGI.pm and the template toolkit. It relies on the jQuery library to allow sortable result tables. The underlying CMCompare algorithm (6) is implemented in Haskell (11). The conversion of input Stockholm-format alignments is done with cmbuild from the Infernal package (3).

The weighted graph representations of the output are created with dot from the graphviz (12) toolset.

The current version of the CMCompare algorithm has a quadratic runtime. With n and m the number of states (roughly the number of columns) in each covariance model, and c a fairly large constant, the runtime is $O(cnm)$. Wall-clock runtimes are from <1s for small models to ≈ 30 s for comparisons between members of the RNaseP clan. We plan to improve on these runtimes in the near future to facilitate large-scale comparisons.

Other tools

To our knowledge, there are no algorithms available other than CMCompare that compare RNA family models with each other. Other classes of biopolymers like DNA or Proteins families can be modeled by profile hidden Markov models (HMMs) (13). General work has been done on comparing HMMs (14) with other HMMs. Also comparisons of HMMs with stochastic context free grammars (15), which provide the underlying principles of covariance models, have been investigated, but in both cases, no available tools originated from this work.

DISCUSSION

CMCws simplifies dealing with an increasing number of RNA family models. Covariance models designed for essentially the same structural RNA family can be detected, as can those that capture a sub- or super-set of the structural features. Covariance models with inferior discriminatory power are easily detected by a large number of high Link scores to other RNA family models. Potential clans can be discovered by looking for a small set of CMs with higher Link scores to each other but low Link scores to all other families.

Challenges remain in identifying the cause of non-specificity among covariance models and how to defuse it. Suggestions how to split RNA families into more specific subfamilies and use of meta-families to pool them again could be a first step into this direction. Also, the construction of clans in an entirely unsupervised manner is a goal for the future.

Promising avenues for expanding functionality of CMCompare are other stochastic grammars such as HMMs used in Pfam (16).

This would allow expanding CMCws in the future to provide a comprehensive web server for comparing and analyzing different kinds of databases of stochastic sequence families.

FUNDING

Austrian FWF, project 'Doktoratskolleg RNA Biology W1207-B09' (to F.E.) and project 'SFB F43 RNA regulation of the transcriptome' (CHzS). Funding for open access charge: Austrian FWF—SFB F43 RNA regulation of the transcriptome.

Conflict of interest statement: None declared.

REFERENCES

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
2. Eddy, S.R. and Durbin, R. (1994) Rna sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.
3. Nawrocki, E.P., Kolbe, D.L. and Eddy, S.R. (2009) Infernal 1.0: inference of RNA alignments. *Bioinformatics*, **25**, 1335–1337.
4. Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A. and Eddy, S.R. (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.

5. Burge,S.W., Daub,J., Eberhardt,R., Tate,J., Barquist,L., Nawrocki,E.P., Eddy,S.R., Gardner,P.P. and Bateman,A. (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.*, **41**, D226–D232.
6. Höner zu Siederdisen,C. and Hofacker,I.L. (2010) Discriminatory power of RNA family models. *Bioinformatics*, **26**, 453–459.
7. Gardner,P.P., Daub,J., Tate,J., Moore,B.L., Osuch,I.H., Griffiths-Jones,S., Finn,R.D., Nawrocki,E.P., Kolbe,D.L., Eddy,S.R. *et al.* (2011) Rfam: wikipedia, clans and the “decimal” release. *Nucleic Acids Res.*, **39**, D141–D145.
8. Lange,S.J., Maticzka,D., Möhl,M., Gagnon,J.N., Brown,C.M. and Backofen,R. (2012) Global or local? Predicting secondary structure and accessibility in mRNAs. *Nucleic Acids Res.*, **40**, 5215–5226.
9. Chen,A. and Brown,C. (2012) Distinct families of cis-acting RNA replication elements epsilon from hepatitis B viruses. *RNA Biol.*, **9**, 1–7.
10. Weinberg,Z., Wang,J.X., Bogue,J., Yang,J., Corbino,K., Moy,R.H. and Breaker,R.R. (2010) Comparative genomics reveals 104 candidate structured RNAs from bacteria, archaea, and their metagenomes. *Genome Biol.*, **11**, R31.
11. The GHC Team. (1989–2013) *The Glasgow Haskell Compiler (GHC)*. <http://www.haskell.org/ghc/>.
12. Ellson,J., Gansner,E., Koutsofios,L., North,S.C. and Woodhull,G. (2002) Graphviz open source graph drawing tools. In: Mutzel,P., Jnger,M. and Leipert,S. (eds), *Graph Drawing*, Vol. 2265 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 483–484.
13. Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.
14. Lyngsø,R.B., Pedersen,C.N. and Nielsen,H. (1999) Metrics and similarity measures for hidden Markov models. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 178–186.
15. Jagota,A., Lyngsø,R.B. and Pedersen,C.N.S. (2001) Comparing a hidden markov model and a stochastic context-free grammar. In: Gascuel,O. and Moret,B.M.E. (eds), *Algorithms in Bioinformatics*, Vol. 2149 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 69–84.
16. Finn,R.D., Mistry,J., Tate,J., Coghill,P., Heger,A., Pollington,J.E., Gavin,O.L., Gunasekaran,P., Ceric,G., Forslund,K. *et al.* (2010) The Pfam protein families database. *Nucleic Acids Res.*, **38**, D211–D222.

8 Discussion and outlook

The goal of this thesis was to facilitate the construction and analysis of *RNA* family models, thereby improving our ability to annotate non-coding *RNA*.

Thousands of *RNA* families for which no *RNA*-family model exists are known alone for *Homo sapiens*. From its first launch, the **Rfam** database has grown at the rate of 176 models on average per year and with 237 models on average per year since **Rfam** version 6.0 (according to **Rfam** ftp-server family files and time stamps, see Figure 2.13). At this rate adding the remaining families for *Homo sapiens* alone will take many years. The tools, described in this thesis aim to facilitate the rapid expansion of the **Rfam** database with novel, high-quality families. To accomplish this, more automatic solutions are needed that reduce manual work required to expand the excellent platform that is already available with **Rfam** (Griffiths-Jones et al., 2003; Gardner et al., 2011; Nawrocki et al., 2014b) and **Infernal** (Nawrocki et al., 2009; Nawrocki and Eddy, 2013).

The importance of adding new *RNA*-family models and making them publicly available is emphasized by the fact that the journal *RNA biology* features a dedicated track (Gardner and Bateman, 2009) for publishing *RNA* family models.

While for many families, automatic construction is feasible, it would be best if the construction process is guided by an expert who specifically knows the respective family. To that end, it is necessary to provide web-enabled tools. A crowd-sourcing effort will be able to fill the gaps that automatic solutions cannot yet cover. The tools developed in this thesis are tailored towards this purpose and either directly provide webservices, or yield results that can be conveniently used as web resources. All tools combined provide a workflow for building *RNA*-family models and are connected with each other as shown in Figure 8.1. The components of the workflow can also be used individually, or even for purposes not connected with *RNA*-family models at all, like **TaxonomyTools**.

The centerpiece of this workflow is **RNAlien**, which constructs *RNA*-family models from a single input-sequence and an optional organism of origin (Eggenhofer et al., 2016). All previously published approaches require a seed align-

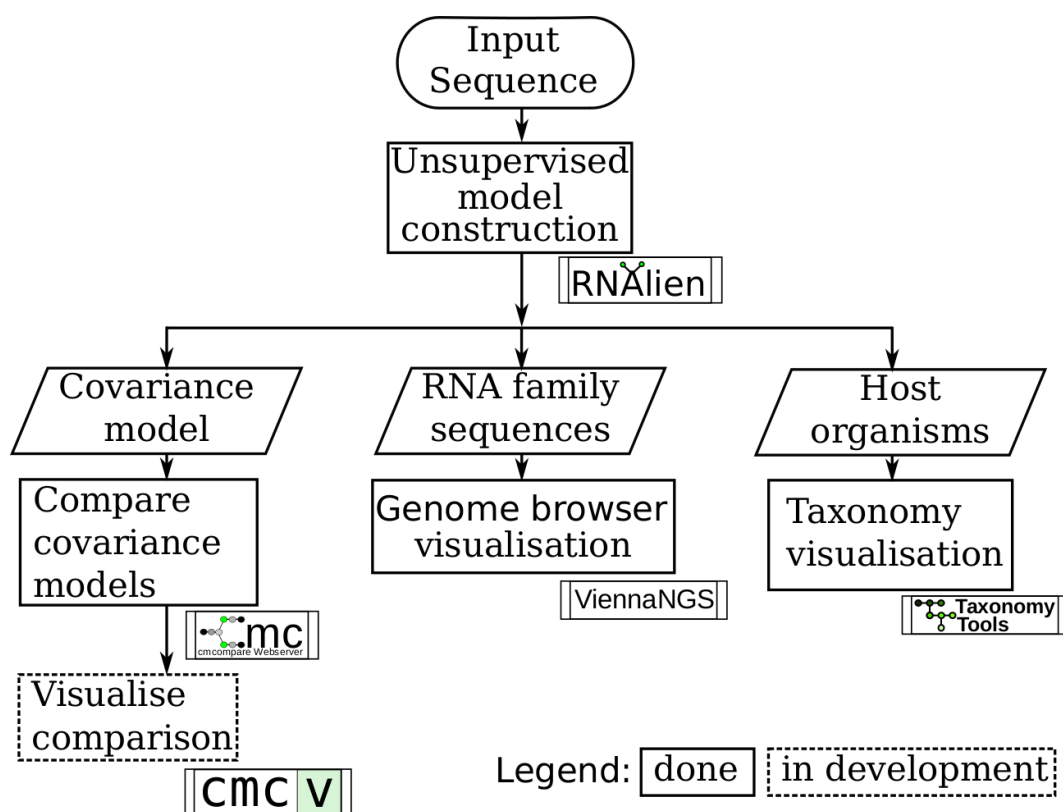


Figure 8.1: Workflow for model construction and evaluation, starting at the top with the construction of a novel *RNA*-family. The workflow branches into three different approaches to evaluate the quality of the model-construction process. The covariance model itself can be evaluated, by **CMCompare** and **cmcv**. The predicted family members can be investigated via genome-browser visualisation in their genomic context. The organisms that host the sequences within an *RNA* family can be inspected as taxonomic tree by **TaxonomyTools**.

ment of representative sequences to build a *RNA*-family, which is then used as a core model to identify further members and is thus extended.

Currently **RNAlien** uses **BLAST** for finding potential family members in the model construction process. **BLAST** allows to offload searches to the **NCBI** infrastructure. Especially for family construction of novel *RNAs* it is relevant that up-to-date sequence databases are searched. **NCBI** is constantly adding new organisms and sequences to their databases, thereby also making these updates available to **RNAlien**. Moreover the user does not have to download the necessary sequence databases, which are of substantial size (e.g *nt* database approximately 28Gb or *refseq-genomic database* approximately 208Gb). Another tool, **nhmmer** should offer a higher sensitivity than **BLAST**, but there is currently no online infrastructure that allows bulk searches with **nhmmer** as

just described. **RNAcentral** offers a **nhmmer** database search for single requests via its web-page, but no interface for automated search requests. It is planned to extend **RNAlien** with a local **nhmmer** search, as well as an online search in the future.

The development of **RNAlien** continues and as a first new feature we have added query soft-masking for the **BLAST** search, that should make searches more profile-like and serve as intermediary step towards **nhmmer**.

Query soft-masking uses conservation information from the alignment to mask weakly conserved nucleotides of the query, such that they are ignored when scoring hits. **RNAlien** uses the conservation information as computed by **cmalign** and passes it to **BLAST**.

The 56 families with known structure already used in the benchmark of the **RNAlien** publication were used here to compare model construction with or without query soft-masking enabled. Specificity was identical in both cases (see Figure 8.2).

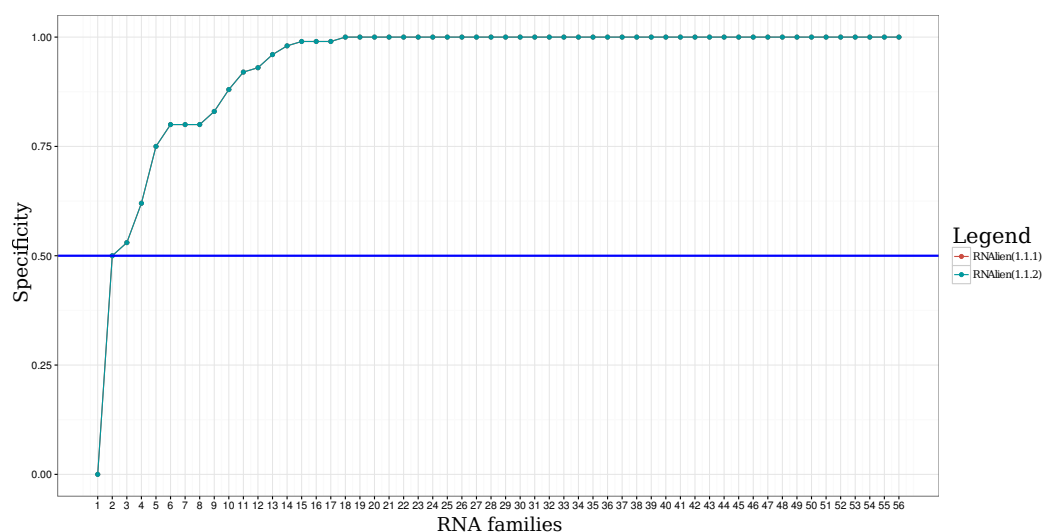


Figure 8.2: Specificity of **RNAlien** with **BLAST** query soft-masking disabled in red (**RNAlien** version 1.1.1) and enabled (**RNAlien** version 1.1.2) in cyan. Both results were identical and both results are overlapping in the plot. The red line is therefore not visible.

Recall was in general lower with soft-masking than without, with the exception of three families (see Figure 8.3). While query soft-masking can be enabled via command-line switch (-f), it is currently not improving results in general. However, now that the technical requirements for using soft-masking with **RNAlien** have been established, further improvements that unlock the full potential of the soft-masking approach can be developed.

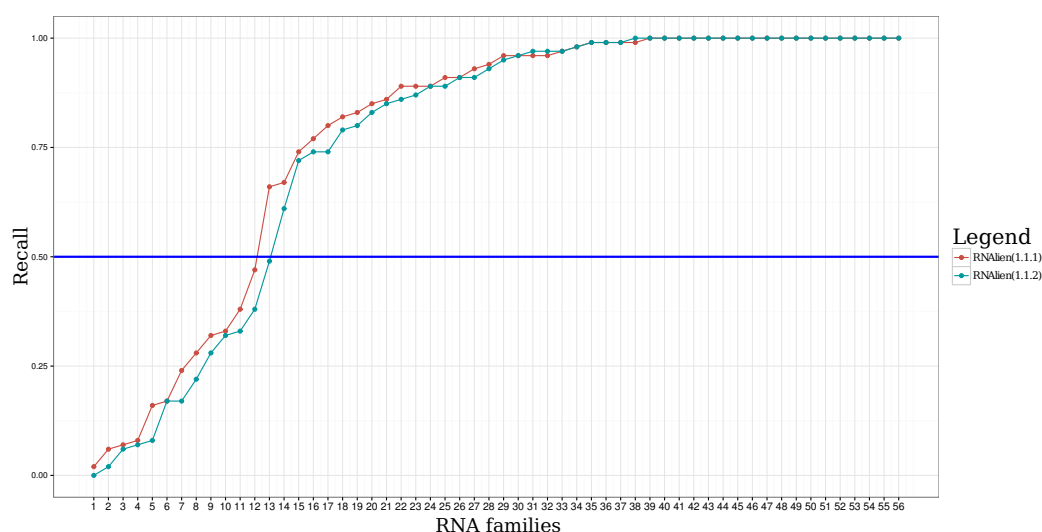


Figure 8.3: Recall of **RNAlien** with BLAST query soft-masking disabled (**RNAlien** version 1.1.1) in red and enabled (**RNAlien** version 1.1.2) in cyan. Recall was in general lower with soft-masking than without, with the exception of three families.

In many cases, the models automatically constructed by **RNAlien** can be used as they are in homology search and in annotation of organisms. There are challenges in the construction process, however, which apply to model construction in general. The first problem affects construction of *RNA*-families that have very distinct subgroups in terms of sequence and structure. In case that these subgroups are connected via gradually different *RNA*-family members it is possible to start in one subgroup and discover the second subgroup via iterative searches. Without *RNA* family members that are similar to both groups, it becomes very difficult to find the second group, even with multiple rounds of searching. However, the finished model should be representative for all real instances of the *RNA*-family, which will not be the case if sequences are missing. These models will have reduced sensitivity in homology search, because they will not be able to detect some family members. The second problem is model construction for *RNA* families with high sequence and structure diversity, which can lead to inclusion of false positive sequences, that are not actually family members. These models will have lower specificity during homology search. An additional problem is that the sequences detected by the model can be truncated compared to the real world situation. While there are families that cannot be constructed entirely automatically, the next step is therefore to support experimentalists and curators in overcoming these challenges. The following tools are designed to identify problems in the model-

construction process and to help an expert for the specific *RNA* family to adjust the construction process accordingly. The model construction process with the workflow can be optimised via several rounds of adjustment and evaluation.

To identify newly constructed family models that are unspecific the **CMCompare webserver** can be used (Eggenhofer et al., 2013). The service therefore compares the model against all against all the models in the **Rfam** database. Unspecific models receive a high link score by **CMCompare**, which means that at least one sequence exists which scores high in both models. While this could indicate a specificity problem in homology search, the compared families could also be members of a *RNA*-family clan. This approach obviously becomes more powerful, the more *RNA* families are available. The **RNAlien** webserver allows to pass the newly constructed model on to the **CMCompare** webserver with a single mouse-click. Besides the evaluation of the *RNA*-family models itself the predicted genomic loci and the corresponding organisms can be investigated.

An expert, which is aware of the functionality of the *RNA*-family can judge the quality of the predictions by comparing them to preexisting annotations in their genomic context. Visualisation of the predictions and annotations via genome- and trackhubs for the **UCSC** genome browser enables to evaluate the loci and their context systematically. However the construction of these hubs is a tedious task. Therefore two tools that provide a simple and efficient way to generate trackhubs and assembly hubs were contributed to the **ViennaNGS** toolkit (Wolfinger et al., 2015).

The taxonomic distribution of detected family members can be useful to evaluate if all instances of the *RNA*-family have been covered. Visualisation of the taxonomic tree of the organisms the *RNA* has been detected in, allows to check if the *RNA* is missing in species where it would be expected, or if the *RNA* was predicted in species where it was not expected. The tools for the simple visualisation and comparison of taxonomic trees are provided in the **TaxonomyTools** package.

High annotation quality for non-coding *RNAs* is of high importance the progress of molecular biology. The workflow and the tools presented in this thesis will contribute to provide new high quality *RNA*-family models, thereby also improving annotation. Specifically the automatic construction should enable the coverage of *RNAs* and organisms that are not in the current focus of research. The results also show that challenging cases in model construction remain that provide open research questions for the future. A part of this gap can be filled by enabling experts for specific *RNA* families to contribute their insights, via

web-enabled tools. Furthermore, there are multiple open avenues to further improve *RNA* family models and the corresponding homology search. Some of these improvements could be integrated directly in the covariance models themselves, while others can be used in combination with the models. Ambivalent *RNA* (Janssen and Giegerich, 2015) family models would offer the possibility to consider multiple secondary structures in the model. This could be very useful in the modeling of *riboswitches* and *RNAs* with multiple stable secondary structures. *RNAs* within a functional context are often located in genomic proximity due to recombination events. Orthology information could therefore be utilized in the evaluation of potential loci. Another promising type of context information are *RNA* 3D modules and motifs. They could also be used to evaluate predicted loci. It is planned to further extend and improve the workflow and its components in the future. The next immediate step is to complete a follow-up project of **CMCompare** (see Figure 8.1) that visualises the regions of the covariance models that were responsible for the detected specificity overlap.

Bibliography

- Alberts, B., Johnson, A., Lewis, J., and et al. (2002). *Molecular Biology of the Cell. 4th edition*. New York: Garland Science.
- Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *Journal of molecular biology*, 219(3):555–565.
- Altschul, S. F. (1993). A protein alignment scoring system sensitive at all evolutionary distances. *Journal of molecular evolution*, 36(3):290–300.
- Altschul, S. F. and Erickson, B. W. (1986). Optimal sequence alignment using affine gap costs. *Bulletin of mathematical biology*, 48(5-6):603–616.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.
- Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.
- Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., et al. (2004). The pfam protein families database. *Nucleic acids research*, 32(suppl 1):D138–D141.
- Baum, L. E. (1972). An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8.
- Ben-Shem, A., de Loubresse, N. G., Melnikov, S., Jenner, L., Yusupova, G., and Yusupov, M. (2011). The structure of the eukaryotic ribosome at 3.0 Å resolution. *Science*, 334(6062):1524–1529.
- Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Wheeler, D. L. (2008). Genbank. *Nucleic acids research*, 36(suppl 1):D25–D30.

- Benton, M. J. (2000). Stems, nodes, crown clades, and rank-free lists: is linnaeus dead? *Biological Reviews of the Cambridge Philosophical Society*, 75(04):633–648.
- Bernhart, S. H., Hofacker, I. L., Will, S., Gruber, A. R., and Stadler, P. F. (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC bioinformatics*, 9(1):1.
- Bock, C. (2012). Analysing and interpreting DNA methylation data. *Nature Reviews Genetics*, 13(10):705–719.
- Bodnar, A., Lichtsteiner, S., Kim, N., Trager, J., et al. (1997). Reconstitution of human telomerase with the template RNA component htr and the catalytic protein subunit hTERT. *Nat Genet*, 17:498–502.
- Brimacombe, R. and Stiege, W. (1985). Structure and function of ribosomal RNA. *Biochemical Journal*, 229(1):1.
- Bringmann, P., Appel, B., Rinke, J., Reuter, R., Theissen, H., and Lührmann, R. (1984). Evidence for the existence of snRNAs u4 and u6 in a single ribonucleoprotein complex and for their association by intermolecular base pairing. *The EMBO journal*, 3(6):1357.
- Brow, D. A. and Guthrie, C. (1988). Spliceosomal RNA u6 is remarkably conserved from yeast to mammals. *Nature*, 334(6179):213–218.
- Browning, D. F. and Busby, S. J. (2004). The regulation of bacterial transcription initiation. *Nature Reviews Microbiology*, 2(1):57–65.
- Brudno, M., Malde, S., Poliakov, A., Do, C. B., Couronne, O., Dubchak, I., and Batzoglou, S. (2003). Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19(suppl 1):i54–i62.
- Burge, S. W., Daub, J., Eberhardt, R., Tate, J., Barquist, L., Nawrocki, E. P., Eddy, S. R., Gardner, P. P., and Bateman, A. (2012). Rfam 11.0: 10 years of RNA families. *Nucleic acids research*, page gks1005.
- Cavalier-Smith, T. (1998). A revised six-kingdom system of life. *Biological Reviews*, 73(3):203–266.
- Cech, T. R. and Steitz, J. A. (2014). The noncoding RNA revolution trashing old rules to forge new ones. *Cell*, 157(1):77–94.

- Cheng, S. and Abelson, J. (1987). Spliceosome assembly in yeast. *Genes & Development*, 1(9):1014–1027.
- Cheong, C. and Moore, P. B. (1992). Solution structure of an unusually stable *RNA* tetraplex containing g-and u-quartet structures. *Biochemistry*, 31(36):8406–8414.
- Chomsky, N. (1959). On certain formal properties of grammars. *Information and control*, 2(2):137–167.
- Cocke, J. (1970). Programming languages and their compilers.
- Comm, I.-I. (1970). Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents. *Biochemistry*, 9(20):4022–4027.
- Coulon, A., Chow, C. C., Singer, R. H., and Larson, D. R. (2013). Eukaryotic transcriptional dynamics: from single molecules to cell populations. *Nature Reviews Genetics*, 14(8):572–584.
- Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.
- Dever, T. E. and Green, R. (2012). The elongation, termination, and recycling phases of translation in eukaryotes. *Cold Spring Harbor perspectives in biology*, 4(7):a013706.
- Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.
- Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9):755–763.
- Eddy, S. R. (2011). Accelerated profile hmm searches. *PLoS Comput Biol*, 7(10):e1002195.
- Eddy, S. R. and Durbin, R. (1994). *RNA* sequence analysis using covariance models. *Nucleic acids research*, 22(11):2079–2088.
- Edgar, R. C. (2004). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797.

- Eggenhofer, F. (2011). *RNApredator*: A web-based tool to predict small *RNA* targets. Master's thesis, University of Vienna.
- Eggenhofer, F., Hofacker, I. L., and Höner zu Siederdissen, C. (2016). *RNAlien* - unsupervised *RNA* family model construction. *Nucleic Acids Research*.
- Eggenhofer, F., Hofacker, I. L., and zu Siederdissen, C. H. (2013). CMCompare webserver: comparing *RNA* families via covariance models. *Nucleic acids research*, 41(W1):W499–W503.
- Eggenhofer, F., Tafer, H., Stadler, P. F., and Hofacker, I. L. (2011). *RNApredator*: fast accessibility-based prediction of *sRNA* targets. *Nucleic Acids Research*, 39(suppl 2):W149–W154.
- Ellis, J. C. and Brown, J. W. (2009). The RNase p family. *RNA biology*, 6(4):362–369.
- Erwig, M. (2001). Inductive graphs and functional graph algorithms. *Journal of Functional Programming*, 11(05):467–492.
- Evans, C. S. and Marler, P. (1994). Food calling and audience effects in male chickens, *gallus gallus*: their relationships to food availability, courtship and social facilitation. *Animal Behaviour*, 47(5):1159–1170.
- Farris, J. S. (1970). Methods for computing wagner trees. *Systematic Biology*, 19(1):83–92.
- Federhen, S. (2012). The NCBI taxonomy database. *Nucleic acids research*, 40(D1):D136–D143.
- Feng, D.-F. and Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of molecular evolution*, 25(4):351–360.
- Field, K. G., Olsen, G. J., Lane, D. J., Giovannoni, S. J., Ghiselin, M. T., Raff, E. C., Pace, N. R., and Raff, R. A. (1988). Molecular phylogeny of the animal kingdom. American Association for the Advancement of Science.
- Finn, R. D., Bateman, A., Clements, J., Coghill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., et al. (2013). Pfam: the protein families database. *Nucleic acids research*, page gkt1223.

- Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416.
- Fitch, W. M. (2000a). Homology: a personal view on some of the problems. *Trends in genetics*, 16(5):227–231.
- Fitch, W. M. (2000b). Homology: a personal view on some of the problems. *Trends in genetics*, 16(5):227–231.
- Fitch, W. M., Margoliash, E., et al. (1967). Construction of phylogenetic trees. *Science*, 155(3760):279–284.
- Frouin, I., Montecucco, A., Spadari, S., and Maga, G. (2003). DNA replication: a complex matter. *EMBO reports*, 4(7):666–670.
- Gan, H. H., Pasquali, S., and Schlick, T. (2003). Exploring the repertoire of *RNA* secondary motifs using graph theory; implications for *RNA* design. *Nucleic acids research*, 31(11):2926–2943.
- Gansner, E. R. and North, S. C. (2000). An open graph visualization system and its applications to software engineering. *SOFTWARE - PRACTICE AND EXPERIENCE*, 30(11):1203–1233.
- Gardner, P. P. and Bateman, A. G. (2009). A home for rna families at rna biology. *RNA biology*, 6(1):2–4.
- Gardner, P. P., Daub, J., Tate, J., Moore, B. L., Osuch, I. H., Griffiths-Jones, S., Finn, R. D., Nawrocki, E. P., Kolbe, D. L., Eddy, S. R., et al. (2011). Rfam: Wikipedia, clans and the “decimal” release. *Nucleic Acids Research*, 39(suppl 1):D141–D145.
- Gaubert, P., Wozencraft, W. C., Cordeiro-Estrela, P., and Veron, G. (2005). Mosaics of convergences and noise in morphological phylogenies: What’s in a viverrid-like carnivoran? *Systematic Biology*, 54(6):865–894.
- Gilbert, W. (1986). Origin of life: The *RNA* world. *Nature*, 319(6055).
- Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351.

- Gorodkin, J., Heyer, L. J., and Stormo, G. D. (1997). Finding common sequence and structure motifs in a set of *RNA* sequences. In *ISMB*, pages 120–123.
- Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of molecular biology*, 162(3):705–708.
- Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13):4355–4358.
- Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S. R. (2003). Rfam: an *RNA* family database. *Nucleic Acids Research*, 31(1):439–441.
- Gruber, A. R., Findeiß, S., Washietl, S., Hofacker, I. L., and Stadler, P. F. (2010). Rnaz 2.0: Improved noncoding *RNA* detection. In *Biocomputing 2010: Proceedings of the Pacific Symposium, Kamuela, Hawaii, USA, 4-8 January 2010*, pages 69–79.
- Gu, X. (2003). Functional divergence in protein (family) sequence evolution. In *Origin and Evolution of New Gene Functions*, pages 133–141. Springer.
- Gumbel, E. (1958). Statistics of extremes. 1958. *Columbia Univ. press, New York*.
- Gyles, C. and Boerlin, P. (2013). Horizontally transferred genetic elements and their role in pathogenesis of bacterial disease. *Veterinary Pathology Online*, page 0300985813511131.
- Harrow, J., Denoeud, F., Frankish, A., Reymond, A., Chen, C.-K., Chrast, J., Lagarde, J., Gilbert, J. G., Storey, R., Swarbreck, D., et al. (2006). GENCODE: producing a reference annotation for ENCODE. *Genome Biol*, 7(1):S4.
- Havgaard, J., Kaur, S., and Gorodkin, J. (2012). Comparative ncRNA gene and structure prediction using foldalign and foldalignm. *Current Protocols in Bioinformatics*, pages 12–11.
- Helm, M. (2006). Post-transcriptional nucleotide modification and alternative folding of *RNA*. *Nucleic acids research*, 34(2):721–733.

- Hendrix, D. K., Brenner, S. E., and Holbrook, S. R. (2005). *RNA structural motifs: building blocks of a modular biomolecule. Quarterly reviews of biophysics*, 38(03):221–243.
- Hermann, H., Fabrizio, P., Raker, V., Foulaki, K., Hornig, H., Brahms, H., and Lührmann, R. (1995). snRNP sm proteins share two evolutionarily conserved sequence motifs which are involved in sm protein-protein interactions. *The EMBO journal*, 14(9):2076.
- Hofacker, I. L., Bernhart, S. H., and Stadler, P. F. (2004). Alignment of *RNA* base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227.
- Hofacker, I. L., Fekete, M., and Stadler, P. F. (2002). Secondary structure prediction for aligned *RNA* sequences. *Journal of molecular biology*, 319(5):1059–1066.
- Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of *RNA* secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188.
- Höner zu Siederdisen, C. and Hofacker, I. L. (2010). Discriminatory power of *RNA* family models. *Bioinformatics*, 26(18):i453–i459.
- Huerta-Cepas, J., Szklarczyk, D., Forslund, K., Cook, H., Heller, D., Walter, M. C., Rattei, T., Mende, D. R., Sunagawa, S., Kuhn, M., et al. (2015). eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic acids research*, page gkv1248.
- Hughey, R. and Krogh, A. (1996). Hidden markov models for sequence analysis: extension and analysis of the basic method. *Computer applications in the biosciences: CABIOS*, 12(2):95–107.
- Janssen, S. and Giegerich, R. (2015). Ambivalent covariance models. *BMC bioinformatics*, 16(1):1.
- Karlin, S. and Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264–2268.
- Kasami, T. (1965). An efficient recognition and syntax analysis algorithm for context-free languages. Technical report, DTIC Document.

- Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002). The human genome browser at UCSC. *Genome research*, 12(6):996–1006.
- Kim, N.-K., Zhang, Q., Zhou, J., Theimer, C. A., Peterson, R. D., and Feigon, J. (2008). Solution structure and dynamics of the wild-type pseudoknot of human telomerase *RNA*. *Journal of molecular biology*, 384(5):1249–1261.
- Knuth, D. E. (1976). Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24.
- Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531.
- Kruger, K., Grabowski, P. J., Zaug, A. J., Sands, J., Gottschling, D. E., and Cech, T. R. (1982). Self-splicing *RNA*: autoexcision and autocyclization of the ribosomal *RNA* intervening sequence of tetrahymena. *cell*, 31(1):147–157.
- Kung, J. T., Colognori, D., and Lee, J. T. (2013). Long noncoding *RNAs*: past, present, and future. *Genetics*, 193(3):651–669.
- Lari, K. and Young, S. J. (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56.
- Lari, K. and Young, S. J. (1991). Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 5(3):237–257.
- Laursen, B. S., Sørensen, H. P., Mortensen, K. K., and Sperling-Petersen, H. U. (2005). Initiation of protein synthesis in bacteria. *Microbiology and Molecular Biology Reviews*, 69(1):101–123.
- Leconte, A. M., Hwang, G. T., Matsuda, S., Capek, P., Hari, Y., and Romesberg, F. E. (2008). Discovery, characterization, and optimization of an unnatural base pair for expansion of the genetic alphabet. *Journal of the American Chemical Society*, 130(7):2336–2343.
- Lee, J. C. and Gutell, R. R. (2004). Diversity of base-pair conformations and their occurrence in *rRNA* structure and *RNA* structural motifs. *Journal of molecular biology*, 344(5):1225–1249.

- Leijen, D. (2001). Parsec, a fast combinator parser.
- Leontis, N. B., Stombaugh, J., and Westhof, E. (2002). The non-watson–crick base pairs and their associated isostericity matrices. *Nucleic acids research*, 30(16):3497–3531.
- Leontis, N. B. and Westhof, E. (2001). Geometric nomenclature and classification of *RNA* base pairs. *Rna*, 7(4):499–512.
- Letunic, I. and Bork, P. (2016). Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic acids research*, page gkw290.
- Linnaeus, C. et al. (1758). *Systema naturae*, vol. 1. *Systema naturae, Vol. 1*.
- Lorenz, R., Bernhart, S. H., Zu Siederdisen, C. H., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):1.
- Maier, L.-K., Alkhnbashi, O. S., Backofen, R., and Marchfelder, A. (2016). CRISPR AND SALTY immune response in haloarchaea. volume RNA metabolism and gene expression in the Archaea of *Nucleic Acids and Molecular Biology*. Springer.
- Makarova, K. S., Wolf, Y. I., Alkhnbashi, O. S., Costa, F., Shah, S. A., Saunders, S. J., Barrangou, R., Brouns, S. J., Charpentier, E., Haft, D. H., et al. (2015). An updated evolutionary classification of CRISPR-cas systems. *Nature Reviews Microbiology*.
- Marabotti, A. and Facchiano, A. (2010). The misuse of terms in scientific literature. *Bioinformatics*, 26(19):2498–2498.
- Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of *RNA* secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(19):7287–7292.
- Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999a). Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *Journal of molecular biology*, 288(5):911–940.

- Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999b). Expanded sequence dependence of thermodynamic parameters improves prediction of *RNA* secondary structure. *Journal of molecular biology*, 288(5):911–940.
- Mathews, D. H. and Turner, D. H. (2002). Dynalign: an algorithm for finding the secondary structure common to two *RNA* sequences. *Journal of molecular biology*, 317(2):191–203.
- Mathews, D. H. and Turner, D. H. (2006). Prediction of *RNA* secondary structure by free energy minimization. *Current opinion in structural biology*, 16(3):270–278.
- Matlin, A. J., Clark, F., and Smith, C. W. (2005). Understanding alternative splicing: towards a cellular code. *Nature reviews Molecular cell biology*, 6(5):386–398.
- McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for *RNA* secondary structure. *Biopolymers*, 29(6-7):1105–1119.
- McWilliam, H., Valentin, F., Goujon, M., Li, W., Narayanasamy, M., Martin, J., Miyar, T., and Lopez, R. (2009). Web services at the european bioinformatics institute-2009. *Nucleic acids research*, 37(suppl 2):W6–W10.
- Michael Bostock, Vadim Ogievetsky, J. H. (2011). D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*.
- Mistry, J., Finn, R. D., Eddy, S. R., Bateman, A., and Punta, M. (2013). Challenges in homology search: Hmmer3 and convergent evolution of coiled-coil regions. *Nucleic acids research*, 41(12):e121–e121.
- Morgenstern, B. (2004). Dialign: multiple dna and protein sequence alignment at bibiserv. *Nucleic acids research*, 32(suppl 2):W33–W36.
- Nawrocki, E. P., Burge, S. W., Bateman, A., Daub, J., Eberhardt, R. Y., Eddy, S. R., Floden, E. W., Gardner, P. P., Jones, T. A., Tate, J., et al. (2014a). Rfam 12.0: updates to the *RNA* families database. *Nucleic acids research*, page gku1063.
- Nawrocki, E. P., Burge, S. W., Bateman, A., Daub, J., Eberhardt, R. Y., Eddy, S. R., Floden, E. W., Gardner, P. P., Jones, T. A., Tate, J., et al.

- (2014b). Rfam 12.0: updates to the *RNA* families database. *Nucleic acids research*, page gku1063.
- Nawrocki, E. P. and Eddy, S. R. (2013). Infernal 1.1: 100-fold faster *RNA* homology searches. *Bioinformatics*, 29(22):2933–2935.
- Nawrocki, E. P., Kolbe, D. L., and Eddy, S. R. (2009). Infernal 1.0: inference of *RNA* alignments. *Bioinformatics*, 25(10):1335–1337.
- Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.
- Notebaart, R. A., Huynen, M. A., Teusink, B., Siezen, R. J., and Snel, B. (2005). Correlation between sequence conservation and the genomic context after gene duplication. *Nucleic acids research*, 33(19):6164–6171.
- Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217.
- Nowakowski, J. and Tinoco, I. (1997). *RNA* structure and stability. In *Seminars in virology*, volume 8, pages 153–165. Elsevier.
- Ohtsuki, T. and Watanabe, Y.-i. (2007). T-armless *tRNAs* and elongated elongation factor tu. *IUBMB life*, 59(2):68–75.
- Palade, G. E. (1955). A small particulate component of the cytoplasm. *The Journal of biophysical and biochemical cytology*, 1(1):59.
- Penny, G. D., Kay, G. F., Sheardown, S. A., Rastan, S., and Brockdorff, N. (1996). Requirement for xist in x chromosome inactivation. *Nature*, 379(6561):131–137.
- Petrov, A. I., Zirbel, C. L., and Leontis, N. B. (2013). Automated classification of *RNA* 3d motifs and the *RNA* 3d motif atlas. *Rna*, 19(10):1327–1340.
- Pruitt, K. D., Tatusova, T., Brown, G. R., and Maglott, D. R. (2012). NCBI reference sequences (refseq): current status, new features and genome annotation policy. *Nucleic acids research*, 40(D1):D130–D135.

- Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., and Glöckner, F. O. (2013). The SILVA ribosomal *RNA* gene database project: improved data processing and web-based tools. *Nucleic acids research*, 41(D1):D590–D596.
- Quigley, G. J. and Rich, A. (1976). Structural domains of transfer *RNA* molecules. *Science*, 194(4267):796–806.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Raghuathan, P. L. and Guthrie, C. (1998). A spliceosomal recycling factor that reanneals u4 and u6 small nuclear ribonucleoprotein particles. *Science*, 279(5352):857–860.
- Raney, B. J., Dreszer, T. R., Barber, G. P., Clawson, H., Fujita, P. A., Wang, T., Nguyen, N., Paten, B., Zweig, A. S., Karolchik, D., et al. (2014). Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC genome browser. *Bioinformatics*, 30(7):1003–1005.
- Reece, J. B., Urry, L. A., Cain, M. L., Wasserman, S. A., Minorsky, P. V., Jackson, R. B., et al. (2011). *Campbell biology*. Pearson Boston.
- Richardson, E. J. and Watson, M. (2013). The automatic annotation of bacterial genomes. *Briefings in bioinformatics*, 14(1):1–12.
- Rivas, E., Clements, J., and Eddy, S. R. (2016). Lack of evidence for conserved secondary structure in long noncoding *RNAs*.
- Robertson, M. P. and Joyce, G. F. (2012). The origins of the *RNA* world. *Cold Spring Harbor perspectives in biology*, 4(5):a003608.
- Rosenbloom, K. R., Armstrong, J., Barber, G. P., Casper, J., Clawson, H., Diekhans, M., Dreszer, T. R., Fujita, P. A., Guruvadoo, L., Haeussler, M., et al. (2015). The UCSC genome browser database: 2015 update. *Nucleic acids research*, 43(D1):D670–D681.
- Sakai, I. (1962). Syntax in universal translation. In *Proc. of the 1961 International Conference on Machine Translation of Languages and Applied Language Analysis*.
- Sankoff, D. (1975). Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28(1):35–42.

- Sankoff, D. (1985). Simultaneous solution of the *RNA* folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825.
- Satish, L. and Gururaj, B. (1993). Use of hidden markov models for partial discharge pattern classification. *Electrical Insulation, IEEE Transactions on*, 28(2):172–182.
- Schroeder, K. T., McPhee, S. A., Ouellet, J., and Lilley, D. M. (2010). A structural database for k-turn motifs in *RNA*. *Rna*, 16(8):1463–1468.
- Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.
- Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical taxonomy. The principles and practice of numerical classification*.
- Sonnhammer, E. L. and Koonin, E. V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *TRENDS in Genetics*, 18(12):619–620.
- Soria, P. S., McGary, K. L., and Rokas, A. (2014). Functional divergence for every paralog. *Molecular biology and evolution*, page msu050.
- Špačková, N. and Šponer, J. (2006). Molecular dynamics simulations of sarcin-ricin r*RNA* motif. *Nucleic acids research*, 34(2):697–708.
- Steijger, T., Abril, J. F., Engström, P. G., Kokocinski, F., Hubbard, T. J., Guigó, R., Harrow, J., Bertone, P., Consortium, R., et al. (2013). Assessment of transcript reconstruction methods for *RNA*-seq. *Nature methods*, 10(12):1177–1184.
- Steinhauer, D. and Holland, J. (1987). Rapid evolution of *RNA* viruses. *Annual Reviews in Microbiology*, 41(1):409–431.
- Storz, G., Vogel, J., and Wassarman, K. M. (2011). Regulation by small *RNAs* in bacteria: expanding frontiers. *Molecular cell*, 43(6):880–891.
- Sundfeld, D., Havgaard, J. H., de Melo, A. C., and Gorodkin, J. (2015). Foldalign 2.5: multithreaded implementation for pairwise structural *RNA* alignment. *Bioinformatics*, page btv748.

- Swiderski, D. L., Zelditch, M. L., and Fink, W. L. (1998). Why morphometrics is not special: coding quantitative data for phylogenetic analysis. *Systematic Biology*, 47(3):508–519.
- Tafer, H., Amman, F., Eggenhofer, F., Stadler, P. F., and Hofacker, I. L. (2011). Fast accessibility-based prediction of *RNA-RNA* interactions. *Bioinformatics*, 27(14):1934–1940.
- Theis, C., Zirbel, C. L., Zu Siederdisen, C. H., Anthon, C., Hofacker, I. L., Nielsen, H., and Gorodkin, J. (2015). *RNA* 3d modules in genome-wide predictions of *RNA* 2d structure. *PloS one*, 10(10):e0139900.
- Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680.
- Torarinsson, E., Havgaard, J. H., and Gorodkin, J. (2007). Multiple structural alignment and clustering of *RNA* sequences. *Bioinformatics*, 23(8):926–932.
- Turner, D. H. and Mathews, D. H. (2009). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic acids research*, page gkp892.
- Urban, J. H. and Vogel, J. (2008). Two seemingly homologous noncoding *RNAs* act hierarchically to activate glms *mRNA* translation. *PLoS Biol*, 6(3):e64.
- Vidovic, I., Nottrott, S., Hartmuth, K., Lührmann, R., and Ficner, R. (2000). Crystal structure of the spliceosomal 15.5 kd protein bound to a u4 sn*RNA* fragment. *Molecular cell*, 6(6):1331–1342.
- Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.
- Wahl, M. C., Will, C. L., and Lührmann, R. (2009). The spliceosome: design principles of a dynamic RNP machine. *Cell*, 136(4):701–718.
- Washietl, S., Hofacker, I. L., and Stadler, P. F. (2005). Fast and reliable prediction of noncoding *RNAs*. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459.

- Watson, J. D., Crick, F. H., et al. (1953). Molecular structure of nucleic acids. *Nature*, 171(4356):737–738.
- Westhof, E. (2010). The amazing world of bacterial structured *RNAs*. *Genome Biol*, 11(3):108.
- Westholm, J. O. and Lai, E. C. (2011). Mirtrons: micro*RNA* biogenesis via splicing. *Biochimie*, 93(11):1897–1904.
- Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Edgar, R., Federhen, S., et al. (2007). Database resources of the national center for biotechnology information. *Nucleic acids research*, 35(suppl 1):D5–D12.
- Wheeler, T. J. and Eddy, S. R. (2013). nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, page btt403.
- Will, C. L. and Lührmann, R. (2011). Spliceosome structure and function. *Cold Spring Harbor perspectives in biology*, 3(7):a003707.
- Will, S., Joshi, T., Hofacker, I. L., Stadler, P. F., and Backofen, R. (2012). LocARNA-p: Accurate boundary prediction and improved detection of structural *RNAs*. *Rna*, 18(5):900–914.
- Will, S., Reiche, K., Hofacker, I. L., Stadler, P. F., and Backofen, R. (2007). Inferring noncoding *RNA* families and classes by means of genome-scale structure-based clustering. *PLoS Comput Biol*, 3(4):e65.
- Woese, C. R., Kandler, O., and Wheelis, M. L. (1990). Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576–4579.
- Wolfinger, M. T., Fallmann, J., Eggenhofer, F., and Amman, F. (2015). ViennaNGS: A toolbox for building efficient next-generation sequencing analysis pipelines. *F1000Research*, 4.
- Yates, A., Akanni, W., Amode, M. R., Barrell, D., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Fitzgerald, S., Gil, L., et al. (2016). Ensembl 2016. *Nucleic acids research*, 44(D1):D710–D716.
- Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3 . *Information and control*, 10(2):189–208.