Grundlagen der Bioinformatik

SoSe 2022

Tutor: Theresa/ Mathias

1	2	3	4	$\sum$

Marina Dittschar & Clarissa Auckenthaler

Blatt 9

(Abgabe am 07.06.2022)

Theoretical Assignments

Task 1: Combinations of hairpins (6)

All hairpin structures for sequences of length n = 4.

The number of hairpin structures for a sequence length of 4 is:

$2^{n-2} - 1 = 2^{4-2} - 1 = 3$

We have drawn all possible combinations in Figure 1.

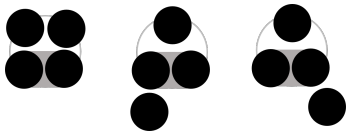

Figure 1: All possible hairpin structures for sequences of length n = 4

Induction

First, we have to determine the number of combinations that is possible for hairpin structures for sequences of a given length. When we have a bases on the one side of the structure and b bases on the other side of the structure, the number of combinations possible with connections that cannot cross is equivalent to choosing k connections between *a-1* bases on the one side and *b-1* bases on the other side [3]. Therefore it holds that [3]:

$$\sum_{k\geq 0}\binom{a-1}{k}\binom{b-1}{k} = \sum_{k\geq 0}\binom{a+b-2}{a-2} = \sum_{k\geq 0}\binom{a+b-2}{b-2}$$

Now because a in the given case is *i-1* and b is n - j, we get [3]:

$$\sum_{k\geq 0}\binom{i-1+n-j-2}{n-j-2}$$

Going on from this, we know that the number of combinations that are possible in a hairpin structure is [3]:

$$\sum_{i=1}^{n-m-1}\frac{n-m-1}{i}$$

We have sequence length n and joining points i with $1 \leq i \leq n$ and j with $i + m + 1 \geq n$
Our loop size m is 1.

Induction start

When we have a sequence of length three, we have
$\sum_{i=1}^{n-2} \binom{n-2}{i} = 2^{n-2} - 1$
$\sum_{i=1}^{1} \binom{3-2}{i} = 2^{3-2} - 1$
$= 1$
possible configurations for a hairpin structure. As you can see in figure 2, this is true.

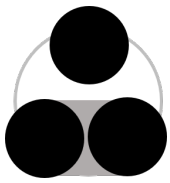

Figure 2: Only possible hairpin structures for sequences of length n = 3

Induction statement

We need to show that

$$\sum_{i=1}^{n+1-2} \frac{n+1-2}{i} = 2^{n+1-2} - 1$$

Induction step

We need to show that if we add another element to the sequence, the difference in the left side of the equation is the same as the difference in the right side of the equation. Firstly, we compute the difference on the right side of equation.
$2^{n+1-2} - 1 - (2^{n-2} - 1)$
$= 2^{n+1-2} - 2^{n-2}$
$= 2^{n-1} - 2^{n-2}$
$= 2^{n-2}$
Therefore, the difference in the left side of the equation has to have this value. The difference on the right side of the equation is, because we take the previous value plus the n+1-value:
$\sum_{i=1}^{n+1-2} \binom{n+1-2}{i}$
$= \sum_{i=1}^{n-1} \binom{n-1}{i}$
$= \sum_{i=1}^{n} \binom{n-2}{i}$
The last step is true, because when we increase the maximum index number of i by one, then we need to decrease the maximum number in the binomial number by one. As we have shown above:

$$\sum_{i=1}^{n} \binom{n-2}{i} = 2^{n-2}$$

This relationship is true because in Pascal's triangle, which is the basis of the binomial coefficient, every element in the current tier is going into the tier below it twice, therefore in every tier the sum doubles. This equals 2^n. Thus, the difference on the two sides is the same. Therefore, the induction is complete.

Task 2: Visualisation of RNA secondary structure (4)

- CACGCUGAACGUACU (hairpin structure)
  Dot-bracket notation:

C	A	C	G	C	U	G	A	A	C	G	U	A	C	U
.	.	.	(	(	.	.	.	.	.	)	)	.	.	.

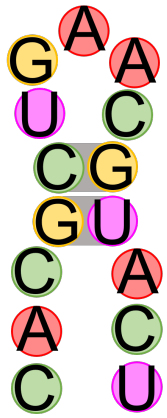

Figure 3: Hairpin

- UCCAGCAGGAAAGC (pseudoknot structure)
  Dot-bracket notation:
  First possibility:

U	C	C	A	G	C	A	G	G	A	A	A	G	C
.	(	.	.	)	(	.	.	.	.	.	.	)	.

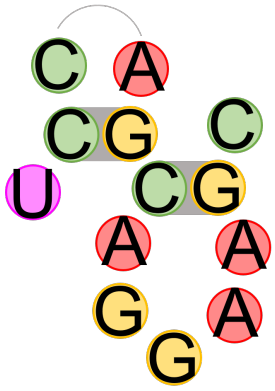

Figure 4: Pseudoknot first possibility

Second Possibility:

U	C	C	A	G	C	A	G	G	A	A	A	G	C
(	(	.	.	.	.	.	.	.	.	.	)	)	.

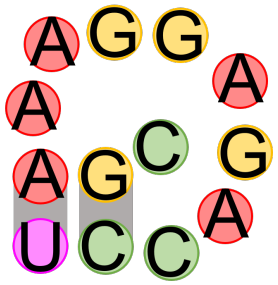

Figure 5: Pseudoknot second possibility

Task 3: Gene prediction

- **First part:** Run PROKKA[1] and GeneMark [2]

  We stored the output from *PROKKA* in a file named:
  `dittschar_auckenthaler_prokka.gff`

  And the output from *GeneMark* in a file named:
  `dittschar_auckenthaler_genemark.gff`

- **Second part:**

  We did not use the provided gff-parser instead we implemented our own gff-parser. The parser is stored in the file named:
  `dittschar_auckenthaler_mygff.py`

  We used Python 3.8.8, and the following libraries **io, pandas, sys, getopt, and numpy**. Additionally we imported our own gff-parser dittschar_auckenthaler_mygff.

  To run the file: `dittschar_auckenthaler_script.py` enter the following code in the command line:
  `python dittschar_auckenthaler_script.py -a dittschar_auckenthaler_prokka.gff -b dittschar_auckenthaler_genemark.gff -r PAO1_annotation.gff`

  Attention: It might take a moment.

  To compare the different files with the ground truth we implemented the following formulas and calculated sensitivity (equation: 1), specificity (equation: 2) and the accuracy (equation: 3) for both tool outcomes. For the formulas we needed the confusion matrix 1:

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} \tag{1}$$

$$\text{Specificity} = \frac{TN}{(TN + FP)} \tag{2}$$

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

		ground truth	
		True	**False**
predicted	**True**	TP	FP
	False	FN	TN

Table 1: Basic Confusion matrix

PROKKA:

		ground truth	
		True	**False**
predicted	**True**	5541520	61785
	False	47241	6878262

Table 2: Confusion matrix: *PROKKA* compared with ground truth

Sensitivity = 0.9915
Specificity = 0.9911
Accuracy = 0.9913

GeneMark:

		ground truth	
		True	**False**
predicted	**True**	5507902	64062
	False	80859	6875985

Table 3: Confusion matrix: *GeneMark* compared with ground truth

Sensitivity = 0.9855
Specificity = 0.9908
Accuracy = 0.9884

Comparison of results:

The accuracy for both models is highly accurate with a value of 98.84% for Genemark and a slightly higher accuracy for *PROKKA* with 99.13%. Generally, both models performed well.

If we have a look at the sensitivity and specificity we can clearly see that *PROKKA* is more sensitive (*PROKKA*: 99.15% and *GeneMark*: 98.55%) and that in terms specificity, both methods approximately lead to the same value, for *PROKKA* the specificity is just minimally higher (99.11%) than for GeneMark (99.08%).

This means that *PROKKA* andd *GeneMark* differ mostly in terms of sensitivity, as *PROKKA* labels more of the true regions correctly. This however did cause a trade-

off in the obtained valuefor specificity. Therefore, in this case, *PROKKA* should be preferred as this method yielded slightly better results.

References

[1] Enis Afgan, Dannon Baker, Bérénice Batut, Marius van den Beek, Dave Bouvier, Martin Cech, John Chilton, Dave Clements, Nate Coraor, Björn A. Grüning, Aysam Guerler, Jennifer Hillman-Jackson, Saskia Hiltemann, Vahid Jalili, Helena Rasche, Nicola Soranzo, Jeremy Goecks, James Taylor, Anton Nekrutenko, and Daniel Blankenberg. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic acids research, 46(W1):W537–W544, 2018.

[2] John Besemer, Alexandre Lomsadze, and Mark Borodovsky. Genemarks: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions. Nucleic acids research, 29 12:2607–18, 2001.

[3] Michael S Waterman. Combinatorics of rna hairpins and cloverleaves. Studies in Applied Mathematics, 60(2):91–98, 1979.