Metagenomic Classification with Deep Learning: Experiments #2

Daniele Bellani

03 • 04 • 2018

Goal

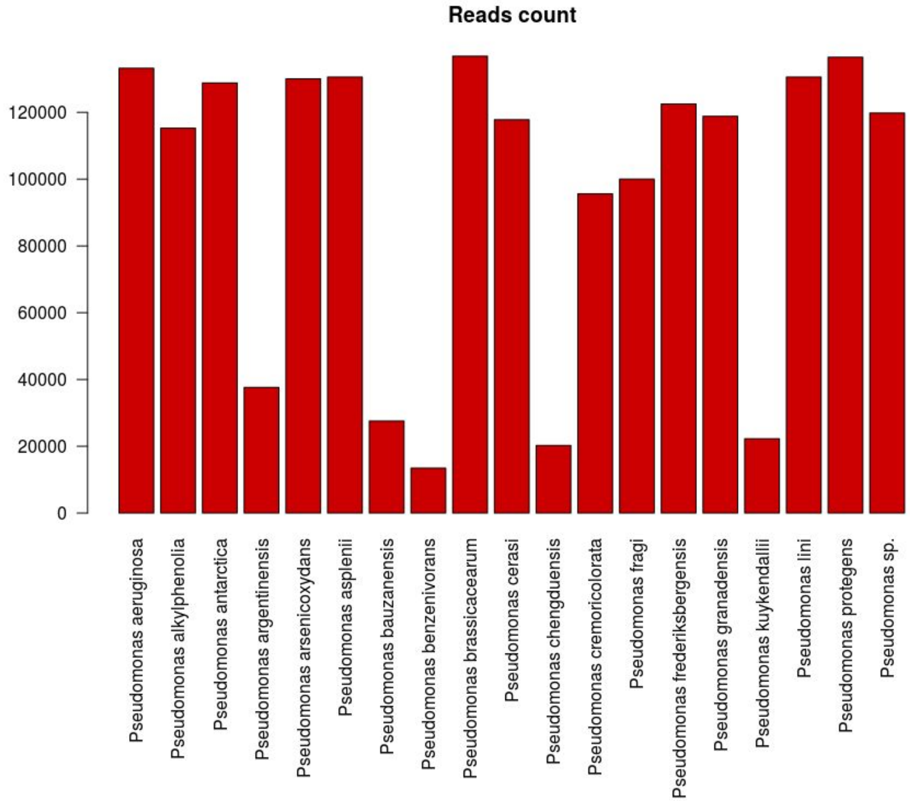
Recognize a specie among others:

one-against-all binary classification

Dataset

- 19 genomes belonging to species of the genus *Pseudomonas*
- Minimum of 90% similarity among them
- Splitting of each genome in 250 base pair **reads**
    - Shifting a cropping window by 50 bp
- Total number of reads per genome = ((GenomeLength-250)/50)+1
- Max number of reads: *Pseudomonas brassicacearum*, 136860
- Min number of reads: *Pseudomonas benzenivorans*, 13459

Dataset reads distribution



Test #1: *Pseudomonas benzenivorans* against all *(least-number-of-reads specie)*

Test 1.1 - Goal

- Study the performances of a chosen network on different training sets
- The training sets differ by composition, i.e. positive/negative examples proportion
- The goal is to study how the different proportions affect the training capabilities of the neural network

Dataset preparation

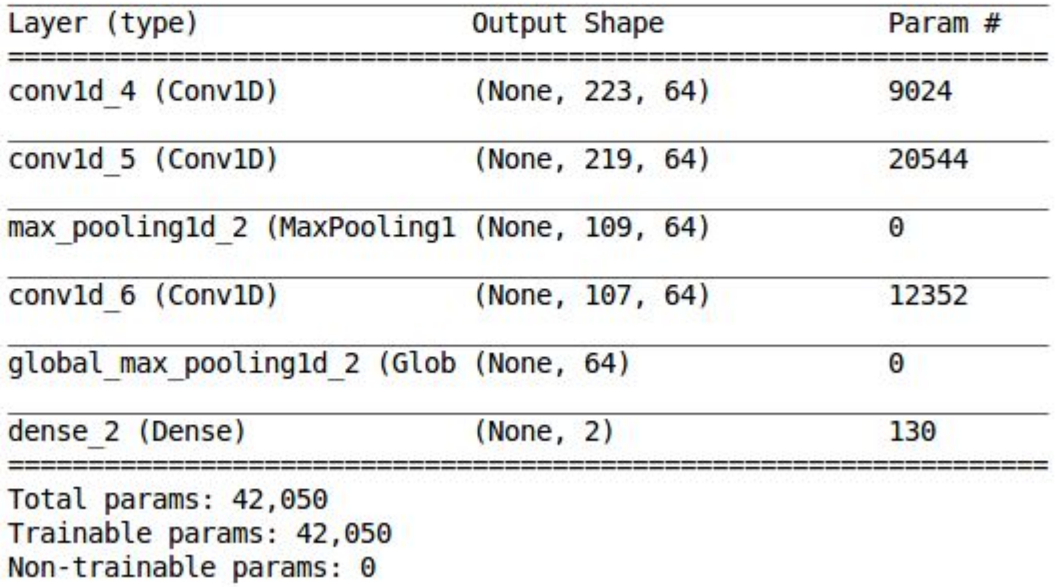
- **Training set**:
  - 80% of positive class reads (Pos = 10767)
  - A number of negative examples proportional to Pos:
    - Neg = Pos*8, Pos*12, Pos*16, Pos*18, Pos*20, Pos*24
- **Test set**:
  - 20% of positive class reads (Pos = 2692)
  - Same amount from all other classes (Neg = Pos*18 = 51130)

Neural network

1. Convolutional(#kernels = 64, kernel_size = 28, activation = ReLU)
2. Convolutional(#kernels = 64, kernel_size = 5, activation = ReLU)
3. MaxPooling(kernel_size = 2)
4. Convolutional(#kernels = 64, kernel_size = 3, activation = ReLU)
5. GlobalMaxPooling(kernel_size = 2)
6. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0005
- Optimizer: Adam
- Epochs: 8*/12*=24, 16*=26, 18*/20*/24*=30
- Batch size: 250

Neural network



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_4 (Conv1D)            (None, 223, 64)           9024
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 219, 64)           20544
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 109, 64)           0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 107, 64)           12352
_________________________________________________________________
global_max_pooling1d_2 (Glob (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 42,050
Trainable params: 42,050
Non-trainable params: 0
```

Training set Neg = Pos*8

- Confusion matrix (holdout):

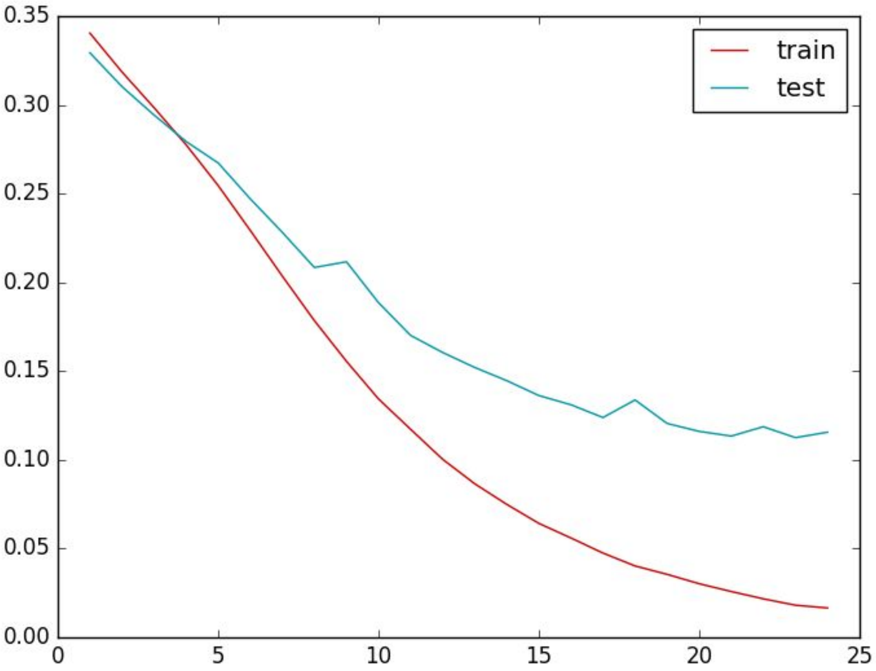
		Predicted	
		Pos	Neg
Real	Pos	2112	580
	Neg	1039	47399

- Precision: 0.670
- Recall: 0.784
- F1: 0.722

Training set Neg = Pos*8

<table>
<tr><th></th><th colspan="2">Fold 1</th><th colspan="2">Fold 2</th><th colspan="2">Fold 3</th><th colspan="2">Fold 4</th><th colspan="2">Fold 5</th></tr>
<tr><td><b>Precision</b></td><td colspan="2">0.829</td><td colspan="2">0.888</td><td colspan="2">0.839</td><td colspan="2">0.849</td><td colspan="2">0.727</td></tr>
<tr><td><b>Recall</b></td><td colspan="2">0.773</td><td colspan="2">0.661</td><td colspan="2">0.811</td><td colspan="2">0.740</td><td colspan="2">0.905</td></tr>
<tr><td><b>F1</b></td><td colspan="2">0.800</td><td colspan="2">0.758</td><td colspan="2">0.825</td><td colspan="2">0.791</td><td colspan="2">0.806</td></tr>
<tr><td rowspan="2"><b>Confusion matrix</b></td><td>2081</td><td>611</td><td>1780</td><td>912</td><td>2185</td><td>507</td><td>1993</td><td>699</td><td>2437</td><td>254</td></tr>
<tr><td>427</td><td>21108</td><td>223</td><td>21312</td><td>419</td><td>21115</td><td>354</td><td>21180</td><td>912</td><td>20622</td></tr>
<tr><td><b>Train loss</b></td><td colspan="2">0.0137</td><td colspan="2">0.0151</td><td colspan="2">0.0169</td><td colspan="2">0.0215</td><td colspan="2">0.0153</td></tr>
<tr><td><b>Test loss</b></td><td colspan="2">0.1120</td><td colspan="2">0.1203</td><td colspan="2">0.1047</td><td colspan="2">0.1113</td><td colspan="2">0.1295</td></tr>
</table>

CV average loss plot (Neg = Pos*8)



Training set Neg = Pos*12

- Confusion matrix (holdout):

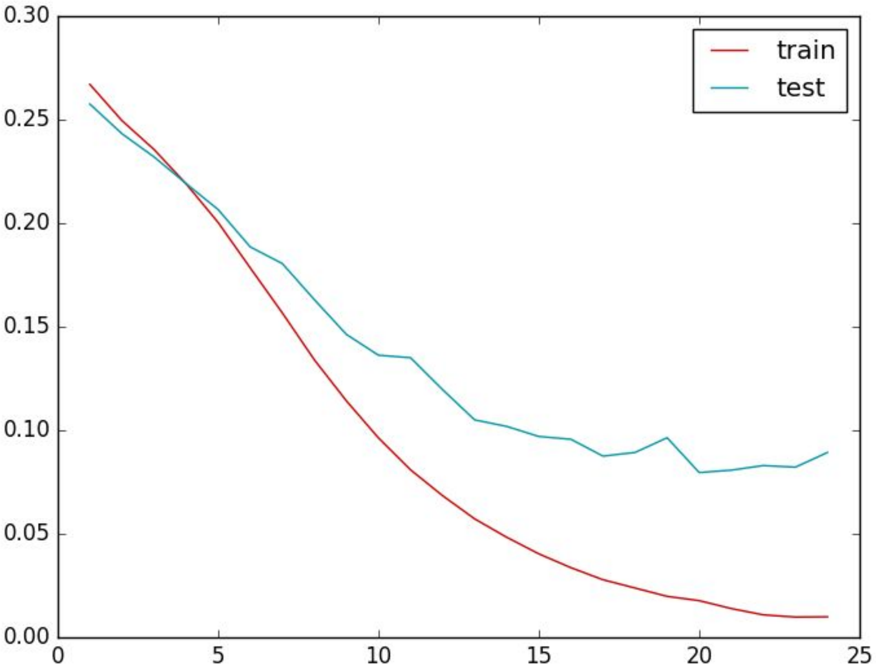
		Predicted	
		Pos	Neg
Real	Pos	2392	300
	Neg	1042	47396

- Precision: 0.696
- Recall: 0.888
- F1: 0.780

Training set Neg = Pos*12

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.808		0.811		0.884		0.667		0.814	
Recall	0.787		0.842		0.690		0.893		0.815	
F1	0.797		0.826		0.775		0.764		0.815	
Confusion matrix	2121	571	2269	423	1859	833	2404	288	2195	496
	504	31798	528	31774	242	32060	1195	31106	499	31802
Train loss	0.0114		0.0080		0.0102		0.0096		0.0102	
Test loss	0.0871		0.0777		0.0900		0.1184		0.0731	

CV average loss plot (Neg = Pos*12)



Training set Neg = Pos*16

- Confusion matrix (holdout):

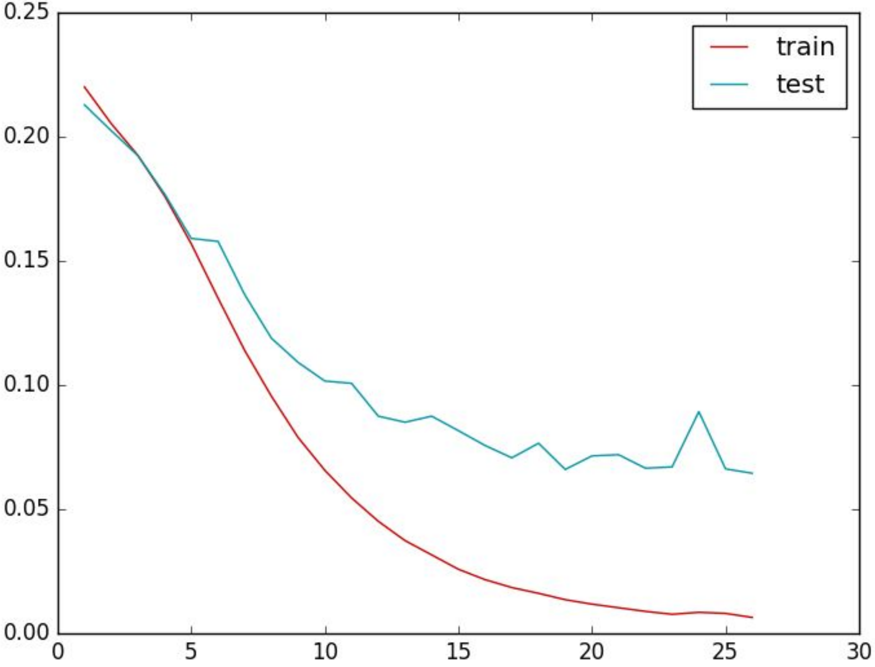
		Predicted	
		Pos	Neg
Real	Pos	1939	753
	Neg	632	47806

- Precision: 0.762
- Recall: 0.845
- F1: 0.801

Training set Neg = Pos*16

<table>
<tr><th></th><th colspan="2">Fold 1</th><th colspan="2">Fold 2</th><th colspan="2">Fold 3</th><th colspan="2">Fold 4</th><th colspan="2">Fold 5</th></tr>
<tr><td>Precision</td><td colspan="2">0.791</td><td colspan="2">0.870</td><td colspan="2">0.771</td><td colspan="2">0.827</td><td colspan="2">0.830</td></tr>
<tr><td>Recall</td><td colspan="2">0.824</td><td colspan="2">0.774</td><td colspan="2">0.859</td><td colspan="2">0.794</td><td colspan="2">0.849</td></tr>
<tr><td>F1</td><td colspan="2">0.807</td><td colspan="2">0.820</td><td colspan="2">0.812</td><td colspan="2">0.810</td><td colspan="2">0.839</td></tr>
<tr><td rowspan="2">Confusion matrix</td><td>2220</td><td>472</td><td>2086</td><td>606</td><td>2313</td><td>379</td><td>2139</td><td>553</td><td>2287</td><td>404</td></tr>
<tr><td>585</td><td>42484</td><td>309</td><td>42760</td><td>686</td><td>42383</td><td>445</td><td>42624</td><td>468</td><td>42600</td></tr>
<tr><td>Train loss</td><td colspan="2">0.0081</td><td colspan="2">0.0041</td><td colspan="2">0.0069</td><td colspan="2">0.0076</td><td colspan="2">0.0053</td></tr>
<tr><td>Test loss</td><td colspan="2">0.0673</td><td colspan="2">0.0587</td><td colspan="2">0.0726</td><td colspan="2">0.0682</td><td colspan="2">0.0554</td></tr>
</table>

CV average loss plot (Neg = Pos*16)



Training set Neg = Pos*18

- Confusion matrix (holdout):

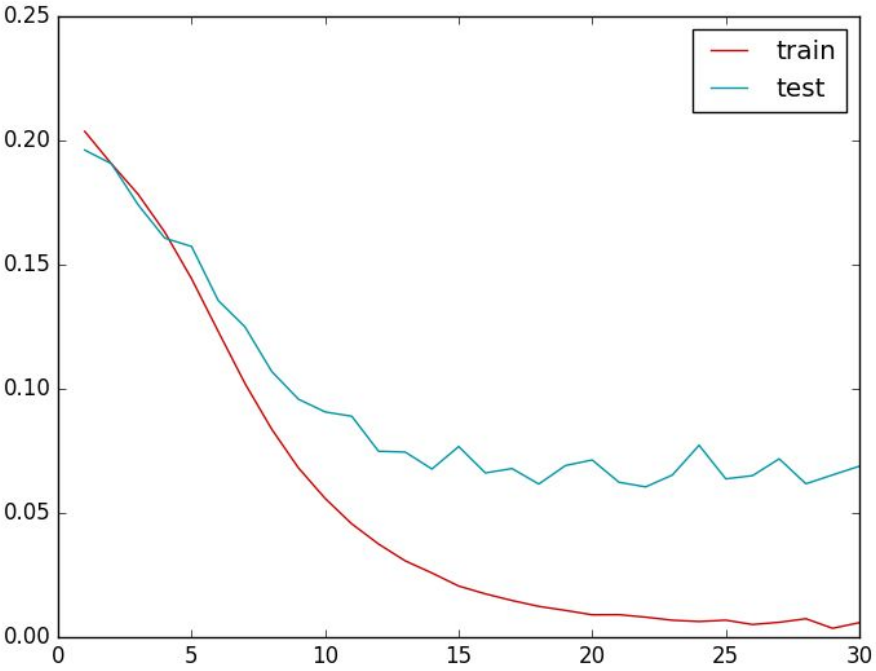
		Predicted	
		Pos	Neg
Real	Pos	1712	980
	Neg	182	48256

- Precision: 0.903
- Recall: 0.635
- F1: 0.746

Training set Neg = Pos*18

<table>
<tr><th></th><th colspan="2">Fold 1</th><th colspan="2">Fold 2</th><th colspan="2">Fold 3</th><th colspan="2">Fold 4</th><th colspan="2">Fold 5</th></tr>
<tr><td>Precision</td><td colspan="2">0.846</td><td colspan="2">0.725</td><td colspan="2">0.866</td><td colspan="2">0.815</td><td colspan="2">0.827</td></tr>
<tr><td>Recall</td><td colspan="2">0.766</td><td colspan="2">0.879</td><td colspan="2">0.702</td><td colspan="2">0.824</td><td colspan="2">0.775</td></tr>
<tr><td>F1</td><td colspan="2">0.804</td><td colspan="2">0.795</td><td colspan="2">0.775</td><td colspan="2">0.820</td><td colspan="2">0.800</td></tr>
<tr><td rowspan="2">Confusion matrix</td><td>2063</td><td>629</td><td>2367</td><td>325</td><td>1891</td><td>801</td><td>2220</td><td>472</td><td>2086</td><td>605</td></tr>
<tr><td>375</td><td>48078</td><td>895</td><td>47558</td><td>292</td><td>48160</td><td>501</td><td>47951</td><td>436</td><td>48016</td></tr>
<tr><td>Train loss</td><td colspan="2">0.0067</td><td colspan="2">0.0029</td><td colspan="2">0.0022</td><td colspan="2">0.0068</td><td colspan="2">0.0104</td></tr>
<tr><td>Test loss</td><td colspan="2">0.0648</td><td colspan="2">0.0777</td><td colspan="2">0.0728</td><td colspan="2">0.0632</td><td colspan="2">0.0653</td></tr>
</table>

CV average loss plot (Neg = Pos*18)



Training set Neg = Pos*20

- Confusion matrix (holdout):

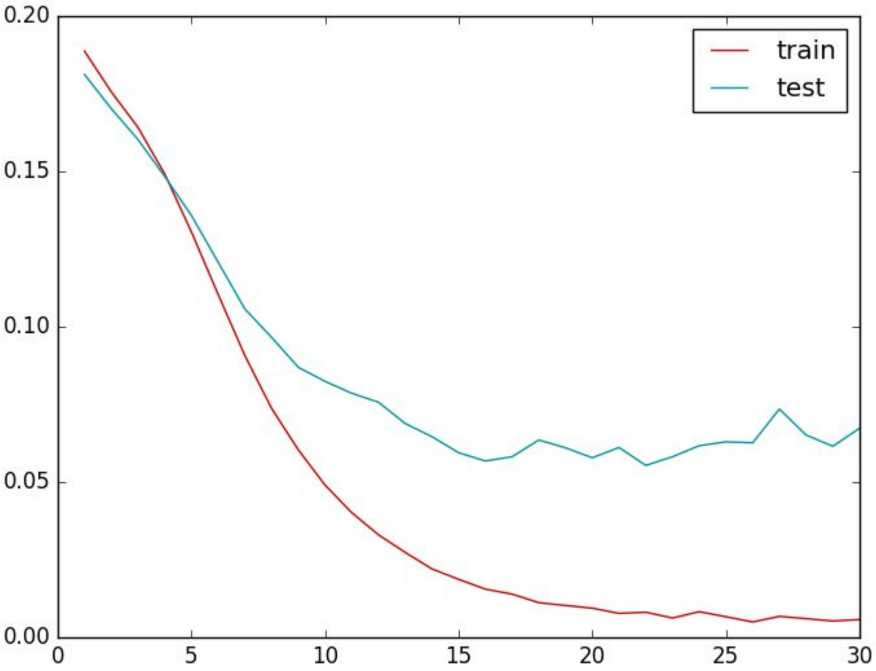
		Predicted	
		Pos	Neg
Real	Pos	2140	552
	Neg	557	47881

- Precision: 0.793
- Recall: 0.794
- F1: 0.794

Training set Neg = Pos*20

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.727		0.798		0.905		0.764		0.798	
Recall	0.813		0.822		0.659		0.823		0.841	
F1	0.768		0.810		0.763		0.792		0.819	
Confusion matrix	2190	502	2215	477	1775	917	2216	476	2265	426
	820	53016	560	53276	185	53651	681	53155	573	53263
Train loss	0.0055		0.0042		0.0059		0.0063		0.0065	
Test loss	0.0831		0.0578		0.0689		0.0718		0.0548	

CV average loss plot (Neg = Pos*20)



Training set Neg = Pos*24

- Confusion matrix (holdout):

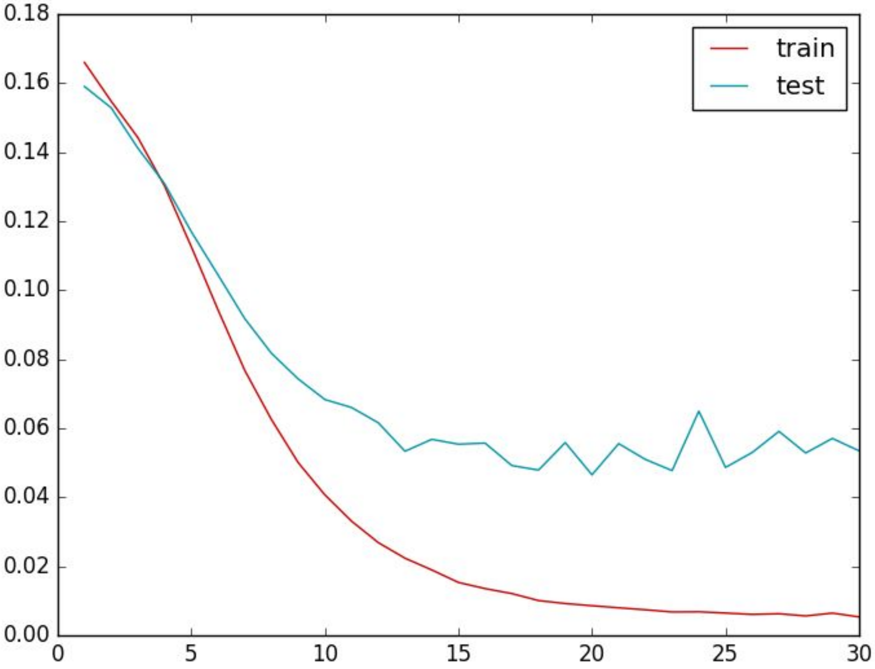
		Predicted	
		Pos	Neg
Real	Pos	2373	319
	Neg	714	47724

- Precision: 0.768
- Recall: 0.881
- F1: 0.821

Training set Neg = Pos*24

<table>
<tr><th></th><th colspan="2">Fold 1</th><th colspan="2">Fold 2</th><th colspan="2">Fold 3</th><th colspan="2">Fold 4</th><th colspan="2">Fold 5</th></tr>
<tr><td>Precision</td><td colspan="2">0.725</td><td colspan="2">0.719</td><td colspan="2">0.887</td><td colspan="2">0.903</td><td colspan="2">0.877</td></tr>
<tr><td>Recall</td><td colspan="2">0.890</td><td colspan="2">0.907</td><td colspan="2">0.723</td><td colspan="2">0.851</td><td colspan="2">0.878</td></tr>
<tr><td>F1</td><td colspan="2">0.799</td><td colspan="2">0.803</td><td colspan="2">0.797</td><td colspan="2">0.876</td><td colspan="2">0.878</td></tr>
<tr><td rowspan="2">Confusion matrix</td><td>2397</td><td>295</td><td>2444</td><td>248</td><td>1949</td><td>743</td><td>2026</td><td>666</td><td>2042</td><td>649</td></tr>
<tr><td>909</td><td>63695</td><td>951</td><td>63652</td><td>248</td><td>64355</td><td>286</td><td>64317</td><td>313</td><td>64290</td></tr>
<tr><td>Train loss</td><td colspan="2">0.0055</td><td colspan="2">0.0052</td><td colspan="2">0.0070</td><td colspan="2">0.0039</td><td colspan="2">0.0047</td></tr>
<tr><td>Test loss</td><td colspan="2">0.0575</td><td colspan="2">0.0559</td><td colspan="2">0.0512</td><td colspan="2">0.0506</td><td colspan="2">0.0518</td></tr>
</table>

CV average loss plot (Neg = Pos*24)



Test 1.2 - Goal

- Changing some of the parameters of the network, in order to improve performances.
- Training set chosen as the one which brought the worst recall (Neg = 18*Pos).

Neural network

1. Convolutional(#kernels = 64, kernel_size = 28, activation = ReLU)
2. Convolutional(#kernels = 96, kernel_size = 5, activation = ReLU)
3. MaxPooling(kernel_size = 2)
4. Convolutional(#kernels = 96, kernel_size = 3, activation = ReLU)
5. GlobalMaxPooling(kernel_size = 2)
6. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0005
- Optimizer: Adam
- Epochs: 18
- Batch size: 250

Kernels: 64-96-96

- Confusion matrix (holdout):

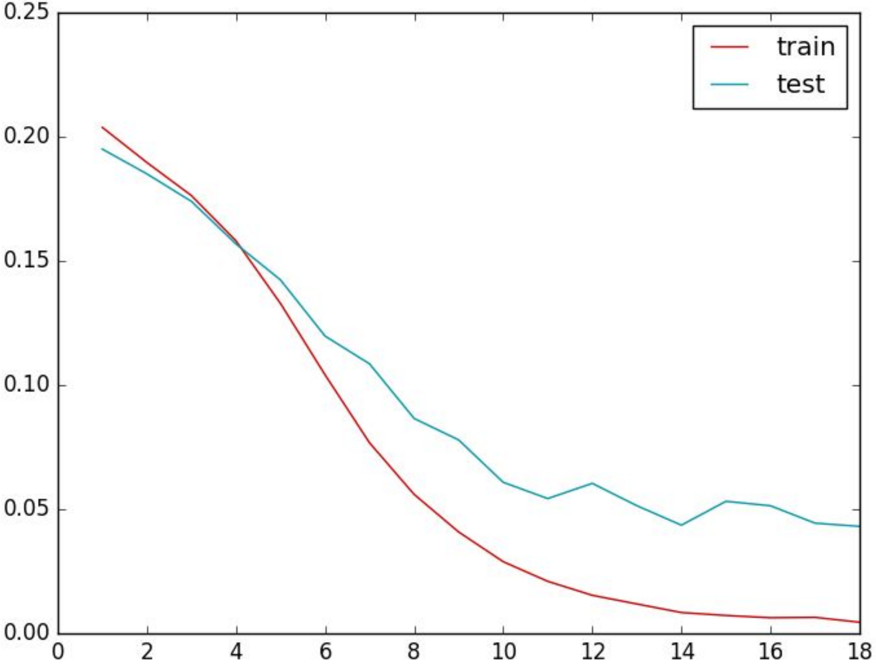
		Predicted	
		Pos	Neg
Real	Pos	2466	226
	Neg	398	48040

- Precision: 0.861
- Recall: 0.916
- F1: 0.887

Kernels: 64-96-96

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.933		0.893		0.846		0.903		0.877	
Recall	0.747		0.819		0.753		0.851		0.878	
F1	0.830		0.855		0.797		0.876		0.878	
Confusion matrix	2013	679	2207	485	2029	663	2292	400	2365	326
	144	48309	263	48190	368	48084	245	48207	331	48121
Train loss	0.0027		0.0052		0.0095		0.0026		0.0023	
Test loss	0.0448		0.0418		0.0553		0.0378		0.0355	

CV average loss plot (64-96-96)



Neural network

1. Convolutional(#kernels = **96**, kernel_size = 28, activation = ReLU)
2. Convolutional(#kernels = **96**, kernel_size = 5, activation = ReLU)
3. MaxPooling(kernel_size = 2)
4. Convolutional(#kernels = **96**, kernel_size = 3, activation = ReLU)
5. GlobalMaxPooling(kernel_size = 2)
6. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0005
- Optimizer: Adam
- Epochs: 22
- Batch size: 250

Kernels: 96-96-96

- Confusion matrix (holdout):

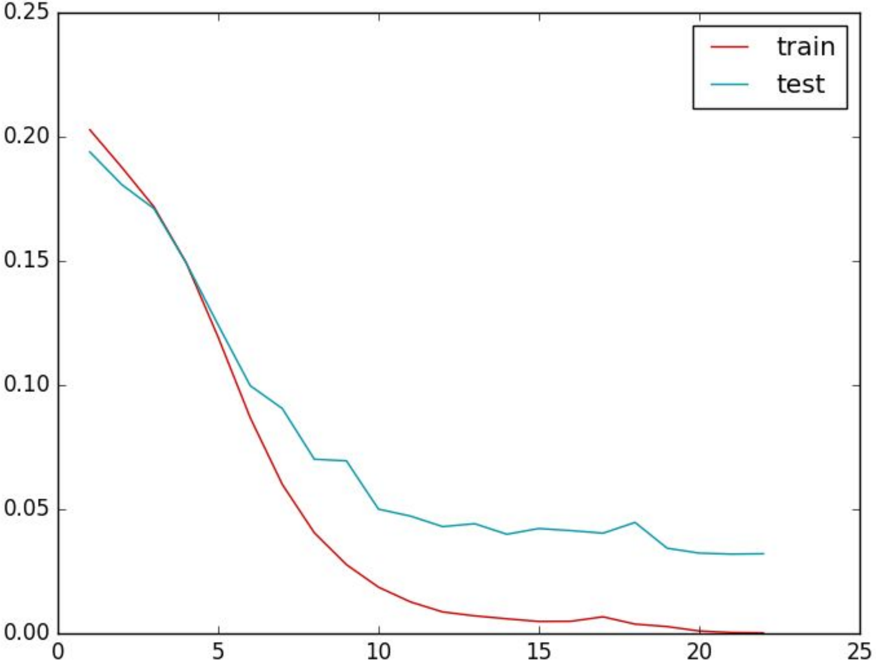
		Predicted	
		Pos	Neg
Real	Pos	2442	250
	Neg	269	48169

- Precision: 0.900
- Recall: 0.907
- F1: 0.903

Kernels: 96-96-96

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.907		0.866		0.922		0.911		0.940	
Recall	0.910		0.953		0.885		0.890		0.879	
F1	0.908		0.908		0.903		0.900		0.909	
Confusion matrix	2451	241	2568	124	2383	309	2398	294	2367	324
	251	48202	395	48058	201	48251	234	48218	149	48303
Train loss	0.0002		0.0001		0.0001		0.0002		0.0002	
Test loss	0.0297		0.0306		0.0363		0.0351		0.0284	

CV average loss plot (96-96-96)



Neural network

1. Convolutional(#kernels = **96**, kernel_size = **21**, activation = ReLU)
2. Convolutional(#kernels = **96**, kernel_size = 5, activation = ReLU)
3. MaxPooling(kernel_size = 2)
4. Convolutional(#kernels = **96**, kernel_size = 3, activation = ReLU)
5. GlobalMaxPooling(kernel_size = 2)
6. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0005
- Optimizer: Adam
- Epochs: 22
- Batch size: 250

Kernels: 96-96-96, Conv1 kernel_size 21

- Confusion matrix (holdout):

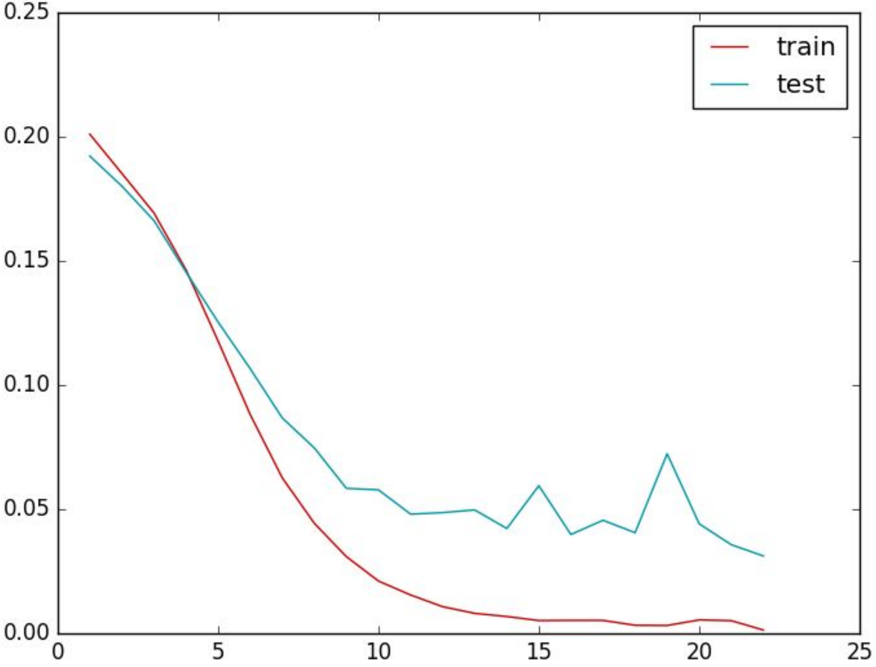
		Predicted	
		Pos	Neg
Real	Pos	2511	181
	Neg	290	48148

- Precision: 0.896
- Recall: 0.932
- F1: 0.914

Kernels: 96-96-96, Conv1 kernel_size 21

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.840		0.923		0.916		0.923		0.889	
Recall	0.952		0.903		0.883		0.895		0.887	
F1	0.892		0.913		0.899		0.909		0.888	
Confusion matrix	2565	127	2433	259	2379	313	2412	280	2388	303
	488	47965	202	48251	218	48234	199	48253	296	48156
Train loss	0.0009		0.0001		0.0009		0.0003		0.0043	
Test loss	0.0337		0.0271		0.0314		0.0295		0.0338	

CV average loss plot (96-96-96, 21)



Neural network

1. Convolutional(#kernels = 96, kernel_size = 35, activation = ReLU)
2. Convolutional(#kernels = 96, kernel_size = 5, activation = ReLU)
3. MaxPooling(kernel_size = 2)
4. Convolutional(#kernels = 96, kernel_size = 3, activation = ReLU)
5. GlobalMaxPooling(kernel_size = 2)
6. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0005
- Optimizer: Adam
- Epochs: 22
- Batch size: 250

Kernels: 96-96-96, Conv1 kernel_size 35

- Confusion matrix (holdout):

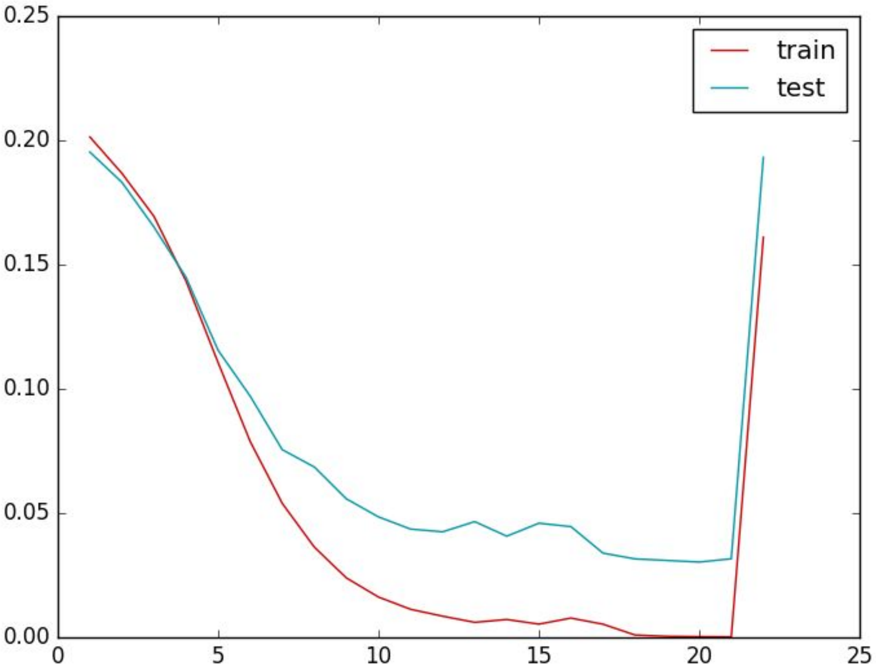
		Predicted	
		Pos	Neg
Real	Pos	2421	271
	Neg	179	48259

- Precision: 0.931
- Recall: 0.899
- F1: 0.914

Kernels: 96-96-96, Conv1 kernel_size 35

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.893		0.910		0		0.900		0.907	
Recall	0.917		0.914		0		0.918		0.923	
F1	0.905		0.912		0		0.909		0.915	
Confusion matrix	2471	221	2463	229	0	2692	2472	220	2484	207
	294	48159	242	48211	0	48452	272	48180	254	48198
Train loss	0.0001		0.0001		0.8043		0.0001		0.0001	
Test loss	0.0304		0.0267		0.8483		0.0325		0.0275	

CV average loss plot (96-96-96, 35)



Neural network

1. Convolutional(#kernels = 128, kernel_size = 28, activation = ReLU)
2. BatchNormalization()
3. Convolutional(#kernels = 128, kernel_size = 5, activation = ReLU)
4. MaxPooling(kernel_size = 2)
5. Convolutional(#kernels = 128, kernel_size = 3, activation = ReLU)
6. GlobalMaxPooling(kernel_size = 2)
7. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0001
- Optimizer: Adam
- Epochs: 21
- Batch size: 250

Kernels: 128-128-128, BatchNorm()

- Confusion matrix (holdout):

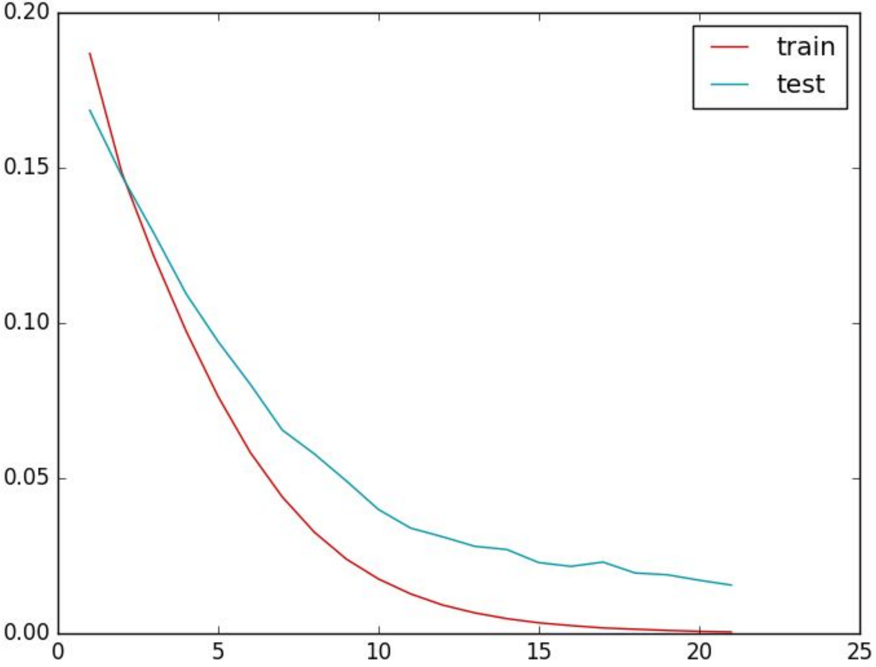
		Predicted	
		Pos	Neg
Real	Pos	2655	37
	Neg	76	48362

- Precision: 0.972
- Recall: 0.986
- F1: 0.979

Kernels: 128-128-128, BatchNorm()

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.967		0.976		0.983		0.971		0.947	
Recall	0.955		0.917		0.916		0.931		0.949	
F1	0.961		0.945		0.948		0.951		0.948	
Confusion matrix	2573	119	2469	223	2466	226	2508	184	2556	135
	86	48367	60	48393	42	48410	74	48378	143	48309
Train loss	0.0003		0.0004		0.0004		0.0004		0.0006	
Test loss	0.0122		0.0155		0.0177		0.0157		0.0164	

CV average loss plot (128-128-128, BN())



Test #2: *Pseudomonas chengduensis* against all *(second least-number-of-reads specie)*

Neural network

1. Convolutional(#kernels = 128, kernel_size = 28, activation = ReLU)
2. BatchNormalization()
3. Convolutional(#kernels = 128, kernel_size = 5, activation = ReLU)
4. MaxPooling(kernel_size = 2)
5. Convolutional(#kernels = 128, kernel_size = 3, activation = ReLU)
6. GlobalMaxPooling(kernel_size = 2)
7. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0001
- Optimizer: Adam
- Epochs: 22
- Batch size: 250

Kernels: 128-128-128, BatchNorm()

- Confusion matrix (holdout):

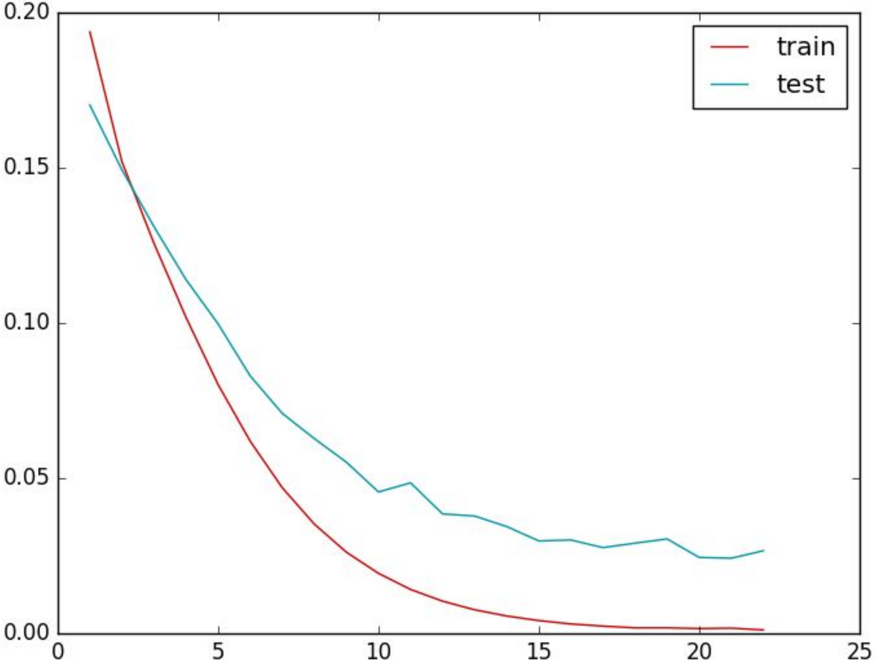
		Predicted	
		Pos	Neg
Real	Pos	3671	381
	Neg	231	72687

- Precision: 0.940
- Recall: 0.905
- F1: 0.923

Kernels: 128-128-128, BatchNorm()

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.975		0.975		0.967		0.854		0.962	
Recall	0.860		0.852		0.884		0.886		0.880	
F1	0.914		0.910		0.924		0.870		0.919	
Confusion matrix	3485	567	2469	596	3584	467	3592	459	3567	484
	86	72836	86	72836	120	72802	610	72311	140	72781
Train loss	0.0004		0.0004		0.0006		0.0032		0.0008	
Test loss	0.0241		0.0252		0.0219		0.0376		0.0240	

CV average loss plot (128-128-128, BN())



Test #3: *Pseudomonas kuykendallii* against all

(third least-number-of-reads specie)

Neural network

1. Convolutional(#kernels = 128, kernel_size = 28, activation = ReLU)
2. BatchNormalization()
3. Convolutional(#kernels = 128, kernel_size = 5, activation = ReLU)
4. MaxPooling(kernel_size = 2)
5. Convolutional(#kernels = 128, kernel_size = 3, activation = ReLU)
6. GlobalMaxPooling(kernel_size = 2)
7. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0001
- Optimizer: Adam
- Epochs: 23
- Batch size: 250

Kernels: 128-128-128, BatchNorm()

- Confusion matrix (holdout):

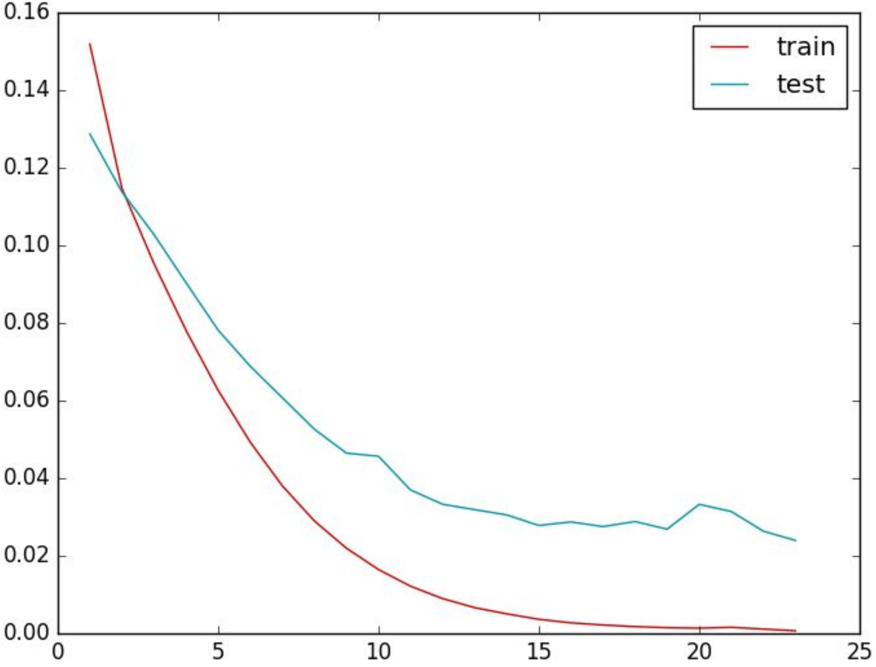
		Predicted	
		Pos	Neg
Real	Pos	4116	335
	Neg	491	79609

- Precision: 0.893
- Recall: 0.924
- F1: 0.908

Kernels: 128-128-128, BatchNorm()

	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Precision	0.927		0.907		0.946		0.933		0.946	
Recall	0.932		0.944		0.866		0.897		0.914	
F1	0.929		0.925		0.904		0.915		0.930	
Confusion matrix	4149	302	4204	247	3857	594	3995	455	4071	379
	325	79786	428	79683	216	79895	284	79827	229	79881
Train loss	0.0004		0.0003		0.0019		0.0003		0.0002	
Test loss	0.0209		0.0239		0.0286		0.0256		0.0206	

CV average loss plot (128-128-128, BN())



Test #4: *Pseudomonas bauzanensis* against all

(fourth least-number-of-reads specie)

Neural network

1. Convolutional(#kernels = 128, kernel_size = 28, activation = ReLU)
2. BatchNormalization()
3. Convolutional(#kernels = 128, kernel_size = 5, activation = ReLU)
4. MaxPooling(kernel_size = 2)
5. Convolutional(#kernels = 128, kernel_size = 3, activation = ReLU)
6. GlobalMaxPooling(kernel_size = 2)
7. Dense(#neurons = 2, activation = Softmax)

- Learning rate: 0.0001
- Optimizer: Adam
- Epochs: 23
- Batch size: 250

Kernels: 128-128-128, BatchNorm()

- Confusion matrix (holdout):

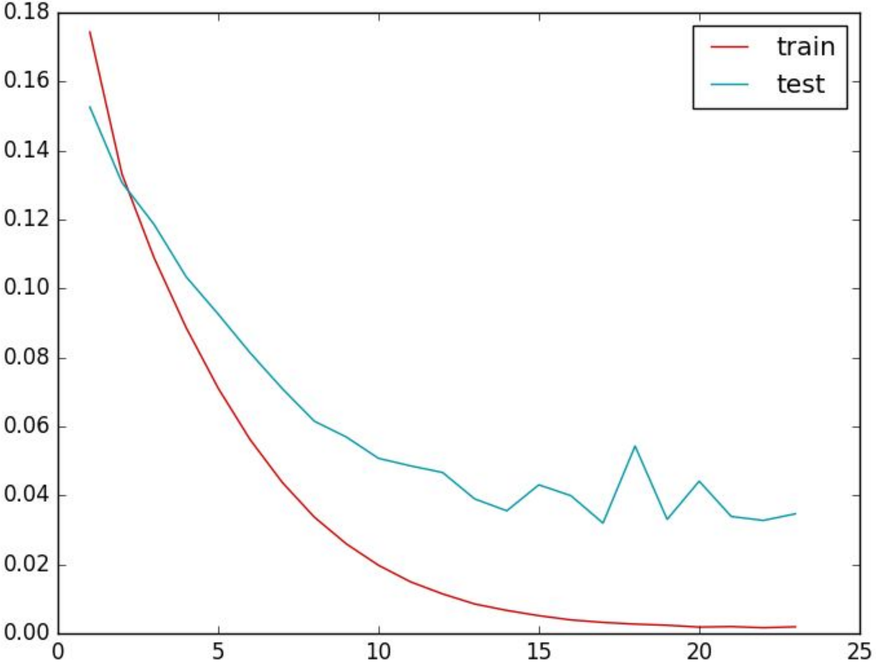
		Predicted	
		Pos	Neg
Real	Pos	5040	477
	Neg	409	98879

- Precision: 0.924
- Recall: 0.913
- F1: 0.919

Kernels: 128-128-128, BatchNorm()

<table>
<tr><th></th><th colspan="2">Fold 1</th><th colspan="2">Fold 2</th><th colspan="2">Fold 3</th><th colspan="2">Fold 4</th><th colspan="2">Fold 5</th></tr>
<tr><td><b>Precision</b></td><td colspan="2">0.923</td><td colspan="2">0.929</td><td colspan="2">0.959</td><td colspan="2">0.865</td><td colspan="2">0.855</td></tr>
<tr><td><b>Recall</b></td><td colspan="2">0.898</td><td colspan="2">0.901</td><td colspan="2">0.737</td><td colspan="2">0.896</td><td colspan="2">0.899</td></tr>
<tr><td><b>F1</b></td><td colspan="2">0.910</td><td colspan="2">0.915</td><td colspan="2">0.834</td><td colspan="2">0.881</td><td colspan="2">0.877</td></tr>
<tr><td rowspan="2"><b>Confusion matrix</b></td><td>4956</td><td>561</td><td>4975</td><td>542</td><td>4071</td><td>1446</td><td>4947</td><td>569</td><td>4964</td><td>552</td></tr>
<tr><td>409</td><td>98890</td><td>378</td><td>98921</td><td>170</td><td>99129</td><td>766</td><td>98533</td><td>837</td><td>98461</td></tr>
<tr><td><b>Train loss</b></td><td colspan="2">0.0007</td><td colspan="2">0.0006</td><td colspan="2">0.0028</td><td colspan="2">0.0027</td><td colspan="2">0.0025</td></tr>
<tr><td><b>Test loss</b></td><td colspan="2">0.0250</td><td colspan="2">0.0258</td><td colspan="2">0.0483</td><td colspan="2">0.0365</td><td colspan="2">0.0375</td></tr>
</table>

CV average loss plot (128-128-128, BN())


train
test