

Motif Discovery from SELEX-seq data

Chaitanya Rastogi, Dahong Liu, and Harmen Bussemaker
Columbia University, New York, NY, USA

1 Background

This package provides a new and efficient implementation of an approach for analyzing SELEX-seq data. The underlying methodology was developed by Todd Riley and Harmen Bussemaker with significant input from Matthew Slattery and Richard Mann, and described in detail in two papers listed in the References section below: Slattery *et al.* (2011) and Riley *et al.* (2014). We request that you cite both papers when you use this software. This tutorial will walk you through a example SELEX analysis using (down-sampled) data from Slattery *et al.* (2011).

2 Installation

Before installing the **SELEX** package, you will need to install **rJava**, which is available from CRAN. If Java is properly installed on your machine, typing

```
install.packages('rJava')
```

should properly install the package. If you encounter any difficulty installing **rJava**, please refer to its documentation at <http://www.rforge.net/rJava/>. You can continue the rest of the tutorial after installing the **SELEX** package.

3 Initializing the SELEX Workspace

Before you load the **SELEX** package, you need to set the maximum Java memory usage limit:

```
> options(java.parameters="-Xmx1500M")
> library(SELEX)
```

Workflow in the SELEX package is centered around the workspace. The workspace consists of samples and data files you are currently working with, and the saved outputs of the various analyses you have run on these data. The workspace has a physical location on disk, and can be configured at any time:

```
> workDir = "./cache/"
> selex.config(workingDir=workDir, maxThreadNumber=4)
```

Before any analyses can be performed, samples must be made ‘visible’ to the current SELEX session, which can be performed with the `selex.loadAnnotation` or `selex.defineSample` commands. Loading a sample lets the current SELEX session know the experimental setup of your data. You are required to provide round, barcode, variable region, and file path information for each sample you load. `selex.defineSample` is convenient when one needs to quickly analyze new data, but the XML-based database used by `selex.loadAnnotation` can be very useful for long-term storage and cataloging of data.

```
> # Extract example data from package, including XML annotation
> exampleFiles = selex.exempladata(workDir)
> # Load all sample files using XML database
> selex.loadAnnotation(exampleFiles[3])
```

You can use `selex.sampleSummary` to see the currently available datasets:

```
> selex.sampleSummary()
```

| | seqName | sampleName | rounds | leftBarcode | rightBarcode |
|---|-------------------------------|------------------------------|--------|-------------|--------------|
| 3 | R0.libraries | R0.barcodeCG | 0 | TGG | CCACGTC |
| 2 | R0.libraries | R0.barcodeGC | 0 | TGG | CCAGCTG |
| 1 | R2.libraries | ExdHox.R2 | 2 | TGG | CCAGCTG |
| | | leftFlank | | | rightFlank |
| 3 | GTTCAGAGTTCTACAGTCCGACGATCTGG | CCACGTCTCGTATGCCGTCTTCTGCTTG | | | |
| 2 | GTTCAGAGTTCTACAGTCCGACGATCTGG | CCAGCTGTCTGATGCCGTCTTCTGCTTG | | | |
| 1 | GTTCAGAGTTCTACAGTCCGACGATCTGG | CCAGCTGTCTGATGCCGTCTTCTGCTTG | | | |
| | seqFile | | | | |

```

3 .\\cache\\R0.fastq.gz
2 .\\cache\\R0.fastq.gz
1 .\\cache\\R2.fastq.gz

```

Sample handles are an easy way to address visible data, and are used by most package functions to allow easy manipulation of your datasets.

```

> r0train = selex.sample(seqName="R0.libraries",
+                         sampleName="R0.barcodeGC", round=0)
> r0test = selex.sample(seqName="R0.libraries",
+                       sampleName="R0.barcodeCG", round=0)
> r2 = selex.sample(seqName="R2.libraries",
+                  sampleName="ExdHox.R2", round=2)

```

At this point, you are ready to analyze your data.

4 Building the Markov Model

In order to properly identify motifs from a SELEX experiment, one needs to be able to characterize the non-randomness of the initial pool of oligomers. This non-randomness can be represented with a Markov model. In order to choose the optimal Markov model, models of various orders will be evaluated, and the one with the greatest cross-validated predictive capability will be chosen. This requires a testing dataset and finding the longest oligonucleotide length k such that all K-mers within this dataset are found at least 100 times. You can find this value using `selex.kmax`:

```

> kmax.value = selex.kmax(sample=r0test)

```

Now, the optimal Markov model can be found, built, and stored:

```

> mm = selex.mm(sample=r0train, order=NA,
+               crossValidationSample=r0test, Kmax=kmax.value)

```

You can use `selex.mmSummary` to see the cross-validated R^2 values of these models:

```

> selex.mmSummary()

```

| | Sample | Order | MarkovModelType | R |
|---|-----------------------------|-------|-----------------|-----------|
| 1 | R0.libraries.R0.barcodeGC.0 | 0 | DIVISION | 0.9800940 |
| 2 | R0.libraries.R0.barcodeGC.0 | 1 | DIVISION | 0.9892230 |
| 3 | R0.libraries.R0.barcodeGC.0 | 2 | DIVISION | 0.9891462 |
| 4 | R0.libraries.R0.barcodeGC.0 | 3 | DIVISION | 0.9889096 |
| 5 | R0.libraries.R0.barcodeGC.0 | 4 | DIVISION | 0.9837154 |

| | CVSample | CVLength |
|---|-----------------------------|----------|
| 1 | R0.libraries.R0.barcodeCG.0 | 5 |
| 2 | R0.libraries.R0.barcodeCG.0 | 5 |
| 3 | R0.libraries.R0.barcodeCG.0 | 5 |
| 4 | R0.libraries.R0.barcodeCG.0 | 5 |
| 5 | R0.libraries.R0.barcodeCG.0 | 5 |



5 Finding the Optimal Motif Length

By observing how concentrated the distribution of later round K-mers frequencies vs. previous round K-mer frequencies becomes, we can find the optimal binding site length. The Kullback-Leibler divergence metric can be used to measure this concentration for a variety of lengths using `selex.infogain`:

```
> selex.infogain(sample=r2,markovModel=mm)
```

`selex.infogainSummary` can be used to view the results:

```
> selex.infogainSummary()[,1:3]
```

| | Sample | K | InformationGain |
|----|--------------------------|----|-----------------|
| 1 | R2.libraries.ExdHox.R2.2 | 2 | 0.1448297 |
| 2 | R2.libraries.ExdHox.R2.2 | 3 | 0.3623032 |
| 3 | R2.libraries.ExdHox.R2.2 | 4 | 0.6695950 |
| 4 | R2.libraries.ExdHox.R2.2 | 5 | 1.0660545 |
| 5 | R2.libraries.ExdHox.R2.2 | 6 | 1.5104582 |
| 6 | R2.libraries.ExdHox.R2.2 | 7 | 1.9133892 |
| 7 | R2.libraries.ExdHox.R2.2 | 8 | 2.2302201 |
| 8 | R2.libraries.ExdHox.R2.2 | 9 | 2.4158743 |
| 9 | R2.libraries.ExdHox.R2.2 | 10 | 2.4143849 |
| 10 | R2.libraries.ExdHox.R2.2 | 11 | 2.0598432 |
| 11 | R2.libraries.ExdHox.R2.2 | 12 | 1.1818942 |
| 12 | R2.libraries.ExdHox.R2.2 | 13 | 0.1063632 |
| 13 | R2.libraries.ExdHox.R2.2 | 14 | 0.0000000 |
| 14 | R2.libraries.ExdHox.R2.2 | 15 | 0.0000000 |
| 15 | R2.libraries.ExdHox.R2.2 | 16 | 0.0000000 |



To see what the K-mer count tables look like for the optimal length, use `selex.counts`:

```
> table = selex.counts(sample=r2, k=optimallength,
+                       markovModel=mm)
> head(table)
```

| | Kmer | ObservedCount | Probability | ExpectedCount |
|---|-----------|---------------|--------------|---------------|
| 1 | ATGATTGAT | 7118 | 7.741514e-06 | 3.093261 |
| 2 | TGATTGATT | 6068 | 9.665983e-06 | 3.862217 |
| 3 | TGATTGATG | 4310 | 7.506314e-06 | 2.999283 |
| 4 | GATTGATTA | 3996 | 7.916402e-06 | 3.163141 |
| 5 | ATCAATCAT | 3870 | 4.642924e-06 | 1.855164 |
| 6 | TTGATTGAT | 3451 | 9.665982e-06 | 3.862217 |

6 Calculating Affinity and Error

With the optimal binding length, you can estimate the affinity and the standard error of the estimate with `selex.affinities` and a Markov Model:

```
> aff = selex.affinities(sample=r2, k=optimalLength,  
+                       markovModel=mm)  
  
> head(aff)[,1:4]
```

| | Kmer | ObservedCount | Probability | ExpectedCount |
|---|-----------|---------------|--------------|---------------|
| 1 | ATGATTGAT | 7118 | 7.741514e-06 | 3.093261 |
| 2 | TGATTGATT | 6068 | 9.665983e-06 | 3.862217 |
| 3 | TGATTGATG | 4310 | 7.506314e-06 | 2.999283 |
| 4 | GATTGATTA | 3996 | 7.916402e-06 | 3.163141 |
| 5 | ATCAATCAT | 3870 | 4.642924e-06 | 1.855164 |
| 6 | TTGATTGAT | 3451 | 9.665982e-06 | 3.862217 |

| | Affinity | SE |
|---|-----------|------------|
| 1 | 1.0000000 | 0.01676239 |
| 2 | 0.8262924 | 0.01500120 |
| 3 | 0.7902404 | 0.01702298 |
| 4 | 0.7409396 | 0.01657620 |
| 5 | 0.9521244 | 0.02164478 |
| 6 | 0.6231369 | 0.01500120 |

References

- [1] Slattery, M.^{*}, Riley, T.^{*}, Liu, P., Abe, N., Gomez-Alcala, P., Dror, I., Zhou, T., Rohs, R.[†], Honig, B.[†], Bussemaker, H.J.[†], and Mann, R.S.[†]. (2011) *Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins*. **Cell** 147:1270–1282. [PMID:22153072]
- [2] Riley, T.R.^{*}, Slattery, M.^{*}, Abe, N., Rastogi, C., Liu, D., Mann, R.S.[†], and Bussemaker, H.J.[†]. (2014) *SELEX-seq: a method for characterizing the complete repertoire of binding site preferences for transcription factor complexes*. **Methods Mol. Biol.** 1196:255–278. [PMID:25151169]