L1000 Peak Deconvolution

Load Function

In[1]:=
```
$libpath = NotebookDirectory[] <> "/dpeak.so";
dpeak = LibraryFunctionLoad[$libpath, "wll_dpeak",
   {{Real, 2, "Constant"}, {Real, 1, "Constant"},
    {Real, 1, "Constant"}, {Real, 2, "Constant"}, {Real, 1, "Constant"}}, {Real, 3}];
```

Test Peak Deconvolution

The mock distribution is a mixture (2:1) of two normal distributions, centered at 7.3 and 9.4.

In[3]:=
```
peaksdist = MixtureDistribution[{2 / 3., 1 / 3.},
   {NormalDistribution[7.3, 0.27], NormalDistribution[9.4, 0.19]}];
```

The background is obtained by all reads in the well in real data, but we use a uniform distribution for simplicity here.

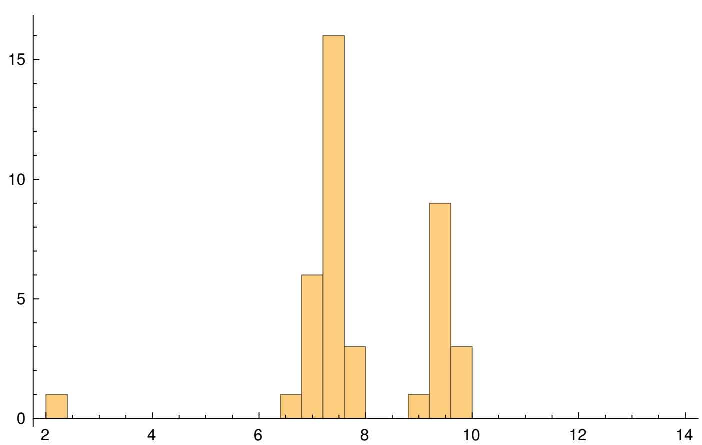
In[4]:=
```
background = UniformDistribution[{2., 14.}];
```

40 reads are sampled from a mixture distribution of the peaks and the background with a misidentification rate (α_c) of 4%.
In the actual calculations, multiple ratios are used, and the likelihood functions are combined by their probability.

In[5]:=
```
reads = RandomVariate[MixtureDistribution[{0.96, 0.04}, {peaksdist, background}], 40];
```

In[6]:=
```
Histogram[reads, {0.4}, PlotRange → {{2, 14}, All}]
```

Out[6]=



Calculate the probability of coming from a misidentified bead at the values of all reads.

In[7]:=
```
bgprob = PDF[background, reads];
```

The grid of all possible peak locations for likelihood calculation.

In[8]:=
```
grid = Range[2.0, 14.0, 0.04];
```

Calculate the log-likelihood function.

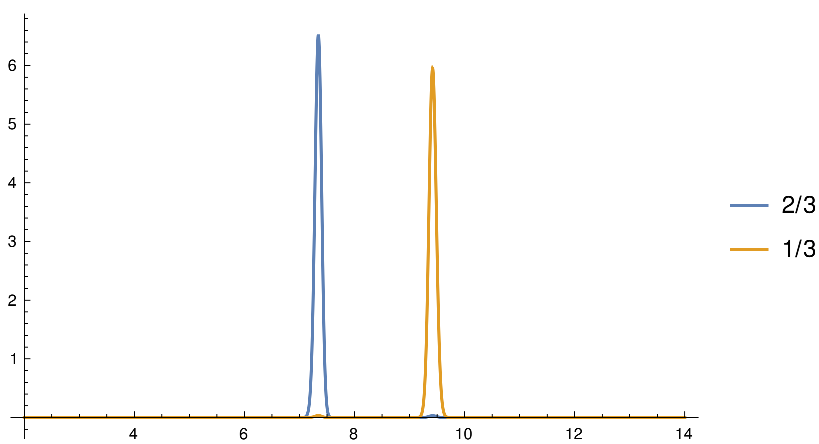
In[9]:=
```
likelihood =
  First@dpeak[{reads}, (*dp52_ratio*){2 / 3.}, (*bg_ratio*){0.04}, {bgprob}, grid];
```

Calculate the marginal distributions of the locations of two peaks.

In[10]:=
```
marginalhi = ProbabilityDistribution[
    Exp@Interpolation[Transpose@
        {grid, Log[# / Total[#] / 0.04] &@Total[Exp@likelihood, {2}]}][x], {x, 2, 14}];
marginallo = ProbabilityDistribution[
    Exp@Interpolation[Transpose@
        {grid, Log[# / Total[#] / 0.04] &@Total[Exp@likelihood, {1}]}][x], {x, 2, 14}];
```

In[12]:=
```
Plot[{PDF[marginalhi, x], PDF[marginallo, x]},
 {x, 2, 14}, PlotRange → All, PlotLegends → {"2/3", "1/3"}]
```

Out[12]=



Scaling/normalization can be added on the marginal distributions.

We choose an idealized reference distribution for the high-abundance peak.

In[13]:=
```
refdist = NormalDistribution[9.0, 1.0];
```

Calculate the z-score based on the quantile of the marginal distribution.

In[14]:=
```
quantilehi = NProbability[x > ref, {x ≈ marginalhi, ref ≈ refdist}];
zscorehi = Sqrt[2] InverseErf[2 quantilehi - 1]
```

Out[15]=
```
-1.62604
```