In [243]:
```
from dcicutils import jh_utils
import cooler
import numpy as np
file_path = jh_utils.mount_4dn_file("4DNFIB59T7NN")
c = cooler.Cooler(file_path + "::/resolutions/100000")
import pandas as pd
import csv
```

In [3]:
```
df = pd.read_csv("diff.csv",sep = ",", header = None)
cols1 = []
cols2 = []
cols3 = []
cols4 = []
for i in range(len(df[0])):
    cols1.append(df[1][i])
    cols2.append(df[2][i])
    cols3.append(df[3][i])
    cols4.append(df[4][i])
print(len(cols1))
```

```
55113
```

In [4]:
```
print(cols1[55112])
print(cols2[55112])
print(cols3[55112])
print(cols4[55112])
```

```
8
37736601
11
8106056
```

In [5]:
```
diff_chr = []
for i in range(len(cols1)):
    A1 = c.matrix().fetch("chr" + str(cols1[i]) + ":" + str(cols2[i]) + "-"
                          "chr" + str(cols3[i]) + ":" + str(cols4[i]) + "-"
    if str(np.log10(A1)[0][0]) != "nan":
        diff_chr.append(np.log10(A1)[0][0])
    print(i)

for j in range(len(diff_chr)):
    print(diff_chr[j])
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

In [256]:
```
print(len(diff_chr))
```

```
54215
```

In [6]:
```
con_diff_chr = []
for i in range(len(diff_chr)):
    con_diff_chr.append(1/diff_chr[i])
print(con_diff_chr)
```

```
[-0.2161663365413809, -0.22567317977981718, -0.20724776765669178, -0.21
539798328850787, -0.22003076548597755, -0.22949452979839388, -0.2161675
4239621616, -0.2203110538746805, -0.23076462231094233, -0.2198594654570
0458, -0.22143641451036522, -0.21062726939368842, -0.21329991761724057,
-0.22559707016709352, -0.20281456243917403, -0.2208204413078103, -0.191
96964182307957, -0.21989582159816076, -0.2177185760641785, -0.232452138
87567998, -0.19434223674297885, -0.2208595887767259, -0.229379891733247
58, -0.21418142018381953, -0.2290391328826558, -0.2184208399667398, -0.
0, -0.0, -0.199650775075996, -0.0, -0.0, -0.21260508199650263, -0.0, -
0.20216470278092688, -0.199650775075996, -0.199650775075996, -0.2120160
2773601996, -0.19268961442830063, -0.23749436690036183, -0.213517057901
9129, -0.2189094792628557, -0.21097453297312366, -0.0, -0.2033173096899
105, -0.0, -0.21995547588849235, -0.20346133717946133, -0.2216052116370
601, -0.21552639279640556, -0.19150221634749295, -0.21733232285640744,
-0.20368988998727727, -0.21428954331430453, -0.22204209264443595, -0.21
56802344419213, -0.21519335438678397, -0.22164529380060963, -0.21424565
300980317, -0.23065415011191026, -0.20400585989340853, -0.2306436434173
2702, -0.22112767443298073, -0.19332038734799795, -0.21248885873330015,
-0.22260884431044456, -0.2229903347312888, -0.2340133422956259, -0.2146
```

In [7]:
```
for i in range(len(con_diff_chr)):
    print(i, con_diff_chr[i])
```

```
0 -0.2161663365413809
1 -0.22567317977981718
2 -0.20724776765669178
3 -0.21539798328850787
4 -0.22003076548597755
5 -0.22949452979839388
6 -0.21616754239621616
7 -0.2203110538746805
8 -0.23076462231094233
9 -0.21985946545700458
10 -0.22143641451036522
11 -0.21062726939368842
12 -0.21329991761724057
13 -0.22559707016709352
14 -0.20281456243917403
15 -0.2208204413078103
16 -0.19196964182307957
17 -0.21989582159816076
18 -0.2177185760641785
```

In [8]:
```
import math
mean = np.mean(con_diff_chr)
std = np.std(con_diff_chr)
print(mean)
print(std)
```

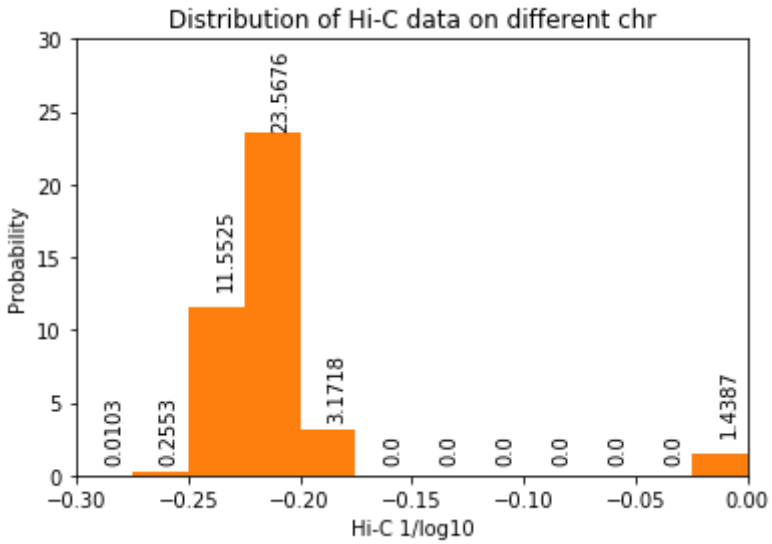
```
-0.21040396850294413
0.042602951015219104
```

In [9]:
```
hcon01 = 0
hcon02 = 0
hcon03 = 0
hcon04 = 0
hcon05 = 0
hcon06 = 0
hcon07 = 0
hcon08 = 0
hcon09 = 0
hcon10 = 0
hcon11 = 0
hcon12 = 0
for i in con_diff_chr:
    if i > -0.025:
        hcon01 += 1
    elif i > -0.05:
        hcon02 += 1
    elif i > -0.075:
        hcon03 +=1
    elif i > -0.1:
        hcon04 +=1
    elif i > -0.125:
        hcon05 +=1
    elif i > -0.15:
        hcon06 +=1
    elif i > -0.175:
        hcon07 +=1
    elif i > -0.2:
        hcon08 +=1
    elif i > -0.225:
        hcon09 +=1
    elif i > -0.25:
        hcon10 +=1
    elif i > -0.275:
        hcon11 +=1
    elif i > -0.3:
        hcon12 +=1

hlb01 = round(hcon01/len(con_diff_chr)/0.025,4)
hlb02 = round(hcon02/len(con_diff_chr)/0.025,4)
hlb03 = round(hcon03/len(con_diff_chr)/0.025,4)
hlb04 = round(hcon04/len(con_diff_chr)/0.025,4)
hlb05 = round(hcon05/len(con_diff_chr)/0.025,4)
hlb06 = round(hcon06/len(con_diff_chr)/0.025,4)
hlb07 = round(hcon07/len(con_diff_chr)/0.025,4)
hlb08 = round(hcon08/len(con_diff_chr)/0.025,4)
hlb09 = round(hcon09/len(con_diff_chr)/0.025,4)
hlb10 = round(hcon10/len(con_diff_chr)/0.025,4)
hlb11 = round(hcon11/len(con_diff_chr)/0.025,4)
hlb12 = round(hcon12/len(con_diff_chr)/0.025,4)
print(hlb01,hlb02,hlb03,hlb04,hlb05,hlb06,hlb07,hlb08,hlb09,hlb10,hlb11,hlb1
```

```
1.4387 0.0 0.0 0.0 0.0 0.0 0.0 3.1718 23.5676 11.5525 0.2553 0.0103
```

In [254]:
```
import matplotlib.pyplot as plt

def normfun(x,mu,sigma):
    pdf = np.exp(-((x - mu)**2)/(2*sigma**2)) / (sigma * np.sqrt(2*np.pi))
    return pdf

x = np.arange(-0.253,-0.168,0.1)
y = normfun(x, mean, std)
plt.plot(x,y)

plt.hist(con_diff_chr, bins=[-0.3, -0.275, -0.25, -0.225, -0.2, -0.175, -0
plt.title('Distribution of Hi-C data on different chr')
plt.xlabel('Hi-C 1/log10')
plt.ylabel('Probability')
plt.xlim((-0.30, 0))
plt.ylim((0, 30))
plt.text(-(0.3+0.275)/2,1,str(hlb12),rotation = 90)
plt.text(-(0.275+0.25)/2,1,str(hlb11),rotation = 90)
plt.text(-(0.25+0.225)/2,13,str(hlb10),rotation = 90)
plt.text(-(0.225+0.2)/2,24,str(hlb09),rotation = 90)
plt.text(-(0.2+0.175)/2,4,str(hlb08),rotation = 90)
plt.text(-(0.175+0.15)/2,1,str(hlb07),rotation = 90)
plt.text(-(0.15+0.125)/2,1,str(hlb06),rotation = 90)
plt.text(-(0.125+0.1)/2,1,str(hlb05),rotation = 90)
plt.text(-(0.1+0.075)/2,1,str(hlb04),rotation = 90)
plt.text(-(0.075+0.05)/2,1,str(hlb03),rotation = 90)
plt.text(-(0.05+0.025)/2,1,str(hlb02),rotation = 90)
plt.text(-(0.025+0)/2,3,str(hlb01),rotation = 90)

plt.show()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:11: Matplotl
ibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed i
n 3.1. Use 'density' instead.
  # This is added back by InteractiveShellApp.init_path()
```



In [13]:
```
a = mean - std
print(a)
```

```
-0.25300691951816323
```

In [233]:
```
new_df = pd.read_csv("diff_chr_nonan.csv",sep = ",", header = None)
new_cols1 = []
for i in range(len(new_df[0])):
    new_cols1.append(new_df[0][i])
```

In [234]:
```
new_cols2 = []
for i in range(len(new_df[0])):
    new_cols2.append(new_df[2][i])
print(new_cols2)
```

```
['9', '1', '1', '6', '1', '3', '20', '3', '3', '3', '20', '3', '3', '1
7', '20', '17', '3', '3', '17', '17', '17', '13', '7', '7', '3', '3',
'3', '1', '7', 'X', 'X', '7', 'X', '7', '7', 'X', '19', '11', '9', 'X',
'9', 'X', 'X', '15', '5', '1', '5', '1', '1', '2', '17', '9', '6', '1
1', '19', '9', '6', '11', '19', '11', '19', '9', '19', '9', '9', '4',
'19', '12', '2', '5', '5', '4', '19', '12', '2', '5', '5', '19', '12',
'2', '5', '5', '12', '2', '5', '5', '2', '5', '5', '5', '5', '6', '2',
'2', '4', '1', '1', '2', '7', '7', '1', '6', '10', '10', '1', '17',
'X', '14', '14', '9', '6', '6', '19', '14', '12', '12', '17', '1', '1
6', '12', '11', 'X', '2', '10', '2', '10', '10', '2', '17', '1', '12',
'1', '12', '12', '19', '19', '17', '17', '9', '9', '9', '2', '9', '2',
'16', '17', '7', '1', '3', '1', '2', '11', '9', '16', '17', '7', '1',
'3', '1', '11', '12', '2', '16', '17', '7', '1', '3', '1', '2', '11',
'12', '16', '17', '7', '1', '3', '1', '11', '12', '17', '7', '1', '3',
'1', '2', '11', '12', '7', '1', '3', '1', '2', '11', '12', '1', '3',
'1', '2', '11', '12', '3', '2', '11', '12', '1', '2', '11', '12', '2',
'11', '12', '11', '12', '12', '16', '2', '2', '15', '2', '5', '1', '1
9', '3', '12', 'X', '14', '17', '7', '3', '3', '12', 'X', '14', '17',
'7', '3', '12', 'X', '14', '17', '7', 'X', '14', '17', '7', '3', '14',
```

In [16]:
```
chrom = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,"X","Y"]
chromLengthIndex = []
chromLength = []

for i in range(len(chrom)):
    A1 = c.matrix().fetch('chr' + str(chrom[i]))
    chromLengthIndex.append(i)
    chromLength.append(len(A1)*100000)
print(chromLengthIndex)
print(type(chromLength[0]))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
0, 21, 22, 23]
<class 'int'>
```

In [181]:
```
import random
randMatchr = np.zeros((54215, 2))

for i in range(len(new_cols1)):
    randMatchr[i][0] = random.randrange(1,25)
for j in range(len(new_cols1)):
    randMatchr[j][1] = random.randrange(1,25)
    if randMatchr[j][1] == randMatchr[j][0]:
        randMatchr[j][1] = random.randrange(1, 25)
for k in range(len(new_cols1)):
    if randMatchr[k][1] == randMatchr[k][0]:
        randMatchr[k][1] = random.randrange(1, 25)
```

In [183]:
```
for i in range(len(new_cols1)):
    if randMatchr[i][1] == randMatchr[i][0]:
        print("error")
        randMatchr[i][1] = random.randrange(1, 25)
    else:
        continue
```

```
error
error
error
```

In [214]:
```
print(randMatchr)
```

```
[[17. 11.]
 [23.  2.]
 [10.  5.]
 ...
 [20. 17.]
 [24. 14.]
 [10. 22.]]
```

In [215]:
```
randMatLen = np.zeros((54215,2))

for i in range(len(new_cols1)):
    for j in range(24):
        if int(randMatchr[i][0]) == chrom[j]:
            print(randMatchr[i][0])
            randMatLen[i][0] = random.randrange(1, chromLength[j] - 100000
```

```
17.0
10.0
13.0
8.0
2.0
13.0
1.0
2.0
19.0
6.0
15.0
12.0
8.0
11.0
14.0
22.0
22.0
21.0
21.0
```

In [216]:
```
for i in range(len(new_cols1)):
    for j in range(24):
        if int(randMatchr[i][1]) == chrom[j]:
            print(randMatchr[i][1])
            randMatLen[i][1] = random.randrange(1, chromLength[j] - 100000
```

```
11.0
2.0
5.0
9.0
1.0
3.0
14.0
10.0
22.0
16.0
3.0
14.0
10.0
14.0
17.0
5.0
10.0
3.0
21.0
```

In [217]:
```
chr_ran = []
chr_ran2 = []
for i in range(len(randMatchr)):
    if int(randMatchr[i][0]) == 23:
        chr_ran.append("X")
    elif int(randMatchr[i][0]) == 24:
        chr_ran.append("Y")
    else:
        chr_ran.append(int(randMatchr[i][0]))

for j in range(len(randMatchr)):
    if int(randMatchr[j][1]) == 23:
        chr_ran2.append("X")
    elif int(randMatchr[j][1]) == 24:
        chr_ran2.append("Y")
    else:
        chr_ran2.append(int(randMatchr[j][1]))
```

In [218]:
```
print(len(randMatchr))
print(len(chr_ran))
print(len(chr_ran2))
```

```
54215
54215
54215
```

In [220]:
```
different_chr = []
for i in range(len(new_cols1)):
    for j in range(2):
        A5 = c.matrix().fetch("chr" + str(chr_ran[i]) + ":" + str(int(rand
                              "chr" + str(chr_ran2[i]) + ":" + str(int(ran
        try:
            if str(np.log10(A5)[0][0]) != "nan":
                different_chr.append(np.log10(A5)[0][0])
        except IndexError as e:
            continue
    print(i+1,"/54214")
```

```
1 /54214
2 /54214
3 /54214
4 /54214
5 /54214
6 /54214
7 /54214
8 /54214
9 /54214
10 /54214
11 /54214
12 /54214
13 /54214
14 /54214
15 /54214
16 /54214
17 /54214
18 /54214
19 /54214
```

In [222]:
```
con_different_chr = []
for i in range(len(different_chr)):
    con_different_chr.append(1/different_chr[i])
```

In [223]:
```
mean3 = np.mean(con_different_chr)
std3 = np.std(con_different_chr)
print(mean3)
print(std3)
```

```
-0.19168473478066222
0.06348819992955902
```

In [224]:
```
diff01 = 0
diff02 = 0
diff03 = 0
diff04 = 0
diff05 = 0
diff06 = 0
diff07 = 0
diff08 = 0
diff09 = 0
diff10 = 0
diff11 = 0
diff12 = 0
for i in con_different_chr:
    if i > -0.025:
        diff01 += 1
    elif i > -0.05:
        diff02 += 1
    elif i > -0.075:
        diff03 +=1
    elif i > -0.1:
        diff04 +=1
    elif i > -0.125:
        diff05 +=1
    elif i > -0.15:
        diff06 +=1
    elif i > -0.175:
        diff07 +=1
    elif i > -0.2:
        diff08 +=1
    elif i > -0.225:
        diff09 +=1
    elif i > -0.25:
        diff10 +=1
    elif i > -0.275:
        diff11 +=1
    elif i > -0.3:
        diff12 +=1

d01 = round(diff01/len(con_different_chr)/0.025,4)
d02 = round(diff02/len(con_different_chr)/0.025,4)
d03 = round(diff03/len(con_different_chr)/0.025,4)
d04 = round(diff04/len(con_different_chr)/0.025,4)
d05 = round(diff05/len(con_different_chr)/0.025,4)
d06 = round(diff06/len(con_different_chr)/0.025,4)
d07 = round(diff07/len(con_different_chr)/0.025,4)
d08 = round(diff08/len(con_different_chr)/0.025,4)
d09 = round(diff09/len(con_different_chr)/0.025,4)
d10 = round(diff10/len(con_different_chr)/0.025,4)
d11 = round(diff11/len(con_different_chr)/0.025,4)
d12 = round(diff12/len(con_different_chr)/0.025,4)
print(d01,d02,d03,d04,d05,d06,d07,d08,d09,d10,d11,d12)
```

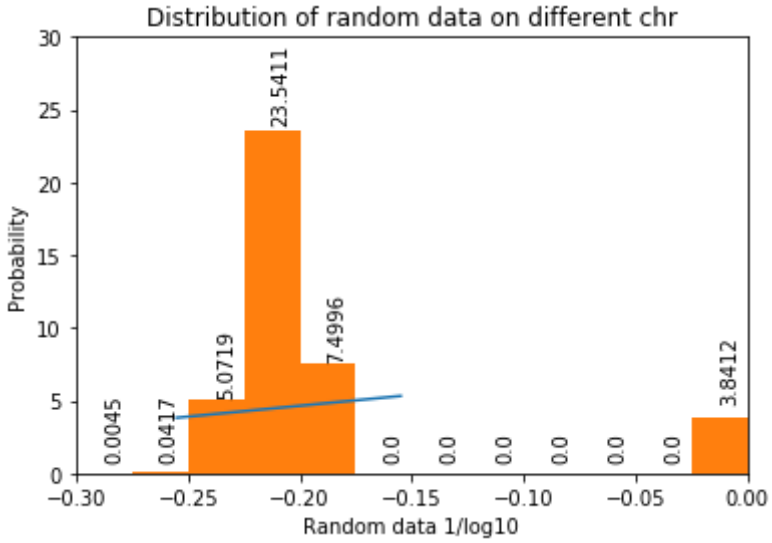
```
3.8412 0.0 0.0 0.0 0.0 0.0 0.0 7.4996 23.5411 5.0719 0.0417 0.0045
```

In [227]:
```
c = mean3 + std3
print(c)
```

```
-0.12819653485110322
```

In [229]:
```
import matplotlib.pyplot as plt
def normfun(x,mu,sigma):
    pdf = np.exp(-((x - mu)**2)/(2*sigma**2)) / (sigma * np.sqrt(2*np.pi))
    return pdf

x3 = np.arange(-0.255,-0.128,0.1)
y3 = normfun(x3, mean3, std3)
plt.plot(x3,y3)

plt.hist(con_different_chr, bins=[-0.3, -0.275, -0.25, -0.225, -0.2, -0.17
plt.title('Distribution of random data on different chr')
plt.xlabel('Random data 1/log10')
plt.ylabel('Probability')
plt.xlim((-0.30, 0))
plt.ylim((0, 30))
plt.text(-(0.3+0.275)/2,1,str(d12),rotation = 90)
plt.text(-(0.275+0.25)/2,1,str(d11),rotation = 90)
plt.text(-(0.25+0.225)/2,5.5,str(d10),rotation = 90)
plt.text(-(0.225+0.2)/2,24.24,str(d09),rotation = 90)
plt.text(-(0.2+0.175)/2,8,str(d08),rotation = 90)
plt.text(-(0.175+0.15)/2,1,str(d07),rotation = 90)
plt.text(-(0.15+0.125)/2,1,str(d06),rotation = 90)
plt.text(-(0.125+0.1)/2,1,str(d05),rotation = 90)
plt.text(-(0.1+0.075)/2,1,str(d04),rotation = 90)
plt.text(-(0.075+0.05)/2,1,str(d03),rotation = 90)
plt.text(-(0.05+0.025)/2,1,str(d02),rotation = 90)
plt.text(-(0.025+0)/2,5,str(d01),rotation = 90)

plt.show()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:10: Matplotl
ibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed i
n 3.1. Use 'density' instead.
  # Remove the CWD from sys.path while we load stuff.
```



In [ ]:

In [235]:
```
import random
randMat1 = np.zeros((54215, 100))
randMat2 = np.zeros((54215, 100))
for i in range(len(new_cols1)):
    for j in range(24):
        if str(new_cols1[i]) == str(chrom[j]):
            for k in range(100):
                randMat1[i][k] = random.randrange(1, chromLength[j] - 1000
    print(i)

for m in range(len(new_cols2)):
    for n in range(24):
        if str(new_cols2[m]) == str(chrom[n]):
            for o in range(100):
                randMat2[m][o] = random.randrange(1, chromLength[n] - 1000
    print(m)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

In [242]:
```
print(randMat2)
```

```
[[2.95129410e+07 1.05760993e+08 1.13119225e+08 ... 4.96476710e+07
  1.72132870e+07 1.07326423e+08]
 [8.58188980e+07 1.27059358e+08 8.38784200e+07 ... 2.24336716e+08
  2.14929291e+08 6.93457590e+07]
 [7.70863800e+07 1.26581693e+08 2.45012150e+08 ... 9.85103300e+07
  6.46237870e+07 1.66775039e+08]
 ...
 [1.06232419e+08 1.19349662e+08 9.37797470e+07 ... 7.41932580e+07
  5.81036570e+07 1.04444286e+08]
 [2.93845440e+07 1.92175380e+07 6.90367600e+06 ... 1.26858870e+07
  9.43213640e+07 7.90203770e+07]
 [1.25151274e+08 4.78091500e+07 6.09810130e+07 ... 3.03741730e+07
  2.20708290e+07 6.39172410e+07]]
```

In [244]:
```
ran_diff_chr = []
for i in range(len(new_cols1)):
    for j in range(100):
        A2 = c.matrix().fetch("chr" + str(new_cols1[i]) + ":" + str(int(ra
                              "chr" + str(new_cols2[i]) + ":" + str(int(ra
        try:
            if str(np.log10(A2)[0][0]) != "nan":
                ran_diff_chr.append(np.log10(A2)[0][0])
        except IndexError as e:
            continue
    print(i+1,"/54214")
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:7: Runtime
Warning: divide by zero encountered in log10
  import sys
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:8: Runtime
Warning: divide by zero encountered in log10
```

In [245]:
```
con_ran_diff_chr = []
for i in range(len(ran_diff_chr)):
    con_ran_diff_chr.append(1/ran_diff_chr[i])
```

In [246]:
```
print(ran_diff_chr[0])
```

```
-5.129315116335726
```

In [247]:
```
mean2 = np.mean(con_ran_diff_chr)
std2 = np.std(con_ran_diff_chr)
print(mean2)
print(std2)
```

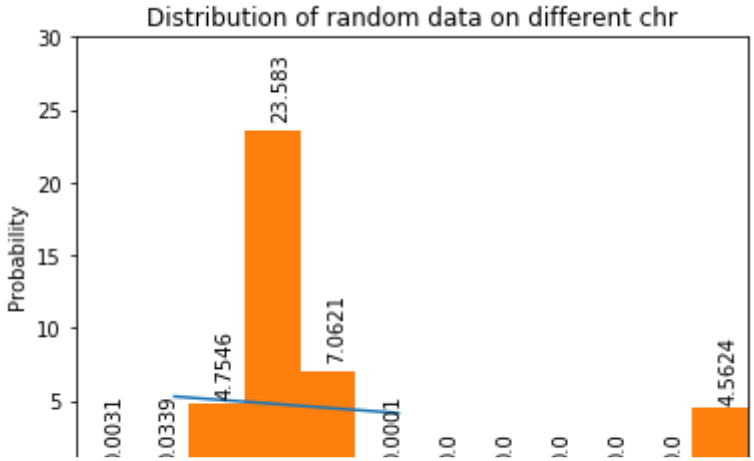
```
-0.18771098092280966
0.06825170298292717
```

In [248]:
```
con01 = 0
con02 = 0
con03 = 0
con04 = 0
con05 = 0
con06 = 0
con07 = 0
con08 = 0
con09 = 0
con10 = 0
con11 = 0
con12 = 0
for i in con_ran_diff_chr:
    if i > -0.025:
        con01 += 1
    elif i > -0.05:
        con02 += 1
    elif i > -0.075:
        con03 +=1
    elif i > -0.1:
        con04 +=1
    elif i > -0.125:
        con05 +=1
    elif i > -0.15:
        con06 +=1
    elif i > -0.175:
        con07 +=1
    elif i > -0.2:
        con08 +=1
    elif i > -0.225:
        con09 +=1
    elif i > -0.25:
        con10 +=1
    elif i > -0.275:
        con11 +=1
    elif i > -0.3:
        con12 +=1

lb01 = round(con01/len(con_ran_diff_chr)/0.025,4)
lb02 = round(con02/len(con_ran_diff_chr)/0.025,4)
lb03 = round(con03/len(con_ran_diff_chr)/0.025,4)
lb04 = round(con04/len(con_ran_diff_chr)/0.025,4)
lb05 = round(con05/len(con_ran_diff_chr)/0.025,4)
lb06 = round(con06/len(con_ran_diff_chr)/0.025,4)
lb07 = round(con07/len(con_ran_diff_chr)/0.025,4)
lb08 = round(con08/len(con_ran_diff_chr)/0.025,4)
lb09 = round(con09/len(con_ran_diff_chr)/0.025,4)
lb10 = round(con10/len(con_ran_diff_chr)/0.025,4)
lb11 = round(con11/len(con_ran_diff_chr)/0.025,4)
lb12 = round(con12/len(con_ran_diff_chr)/0.025,4)
print(lb01,lb02,lb03,lb04,lb05,lb06,lb07,lb08,lb09,lb10,lb11,lb12)
```

```
4.5624 0.0 0.0 0.0 0.0 0.0 0.0001 7.0621 23.583 4.7546 0.0339 0.0031
```

In [255]:
```
import matplotlib.pyplot as plt
def normfun(x,mu,sigma):
    pdf = np.exp(-((x - mu)**2)/(2*sigma**2)) / (sigma * np.sqrt(2*np.pi))
    return pdf

x = np.arange(-0.256,-0.119,0.1)
y = normfun(x, mean, std)
plt.plot(x,y)

plt.hist(con_ran_diff_chr, bins=[-0.3, -0.275, -0.25, -0.225, -0.2, -0.175
plt.title('Distribution of random data on different chr')
plt.xlabel('Random data 1/log10')
plt.ylabel('Probability')
plt.xlim((-0.30, 0))
plt.ylim((0, 30))
plt.text(-(0.3+0.275)/2,1,str(lb12),rotation = 90)
plt.text(-(0.275+0.25)/2,1,str(lb11),rotation = 90)
plt.text(-(0.25+0.225)/2,5.5,str(lb10),rotation = 90)
plt.text(-(0.225+0.2)/2,24.5,str(lb09),rotation = 90)
plt.text(-(0.2+0.175)/2,8,str(lb08),rotation = 90)
plt.text(-(0.175+0.15)/2,1,str(lb07),rotation = 90)
plt.text(-(0.15+0.125)/2,1,str(lb06),rotation = 90)
plt.text(-(0.125+0.1)/2,1,str(lb05),rotation = 90)
plt.text(-(0.1+0.075)/2,1,str(lb04),rotation = 90)
plt.text(-(0.075+0.05)/2,1,str(lb03),rotation = 90)
plt.text(-(0.05+0.025)/2,1,str(lb02),rotation = 90)
plt.text(-(0.025+0)/2,5,str(lb01),rotation = 90)

plt.show()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:10: Matplo
tlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed
in 3.1. Use 'density' instead.
  # Remove the CWD from sys.path while we load stuff.
```



In [251]:
```
a = mean2 + std2
print(a)
```

```
-0.1194592779398825
```

In [ ]:
```

```