Number formatting

In[1]:=
```
number2Printed[number_] := Module[{returnedString = "e", foo, bar, idx, oom},
                                  模組
  If[number == 1, Return["1.0e+00"], If[number == 0, Return["0.0e+00"],
  如果              返回                 如果              返回
    If[number < 1,
    如果
      For[idx = 1, StringLength[returnedString] == 1, idx = idx + 1,
      For迴圈       字串長度
       foo = Floor[number / 10^(-idx)];
             弱取整
       If[foo == 0, ,
       如果
        bar = Round[(number - foo * 10^(-idx)) / 10^(-idx - 1)];
              捨入
        If[StringLength[ToString[idx]] == 1,
        ··· 字串長度          轉換成字串
          returnedString = StringJoin[ToString[foo],
                                      字串結合      轉換成字串
            ".", ToString[bar], returnedString, "-0", ToString[idx]],
                 轉換成字串                                轉換成字串
          returnedString = StringJoin[ToString[foo], ".", ToString[bar],
                                      字串結合      轉換成字串              轉換成字串
            returnedString, "-", ToString[idx]]
                                           轉換成字串
        ]
       ]
      ];
      Return[returnedString]
      返回
      ,
      oom = (StringLength[ToString[DecimalForm[Floor[number] * 1.]]] - 2);
             字串長度          轉換成字串 十進位形式      弱取整
      foo = Floor[number / 10^oom];
            弱取整
      bar = Round[(number - foo * 10^oom) / 10^(oom - 1)];
            捨入
      If[StringLength[ToString[oom]] == 1,
      ··· 字串長度          轉換成字串
        returnedString = StringJoin[ToString[foo],
                                    字串結合      轉換成字串
          ".", ToString[bar], returnedString, "+0", ToString[oom]],
               轉換成字串                                轉換成字串
```

```
        returnedString = StringJoin[ToString[foo], ".", ToString[bar],
                                    字串結合    轉換成字串    轉換成字串
          returnedString, "+", ToString[oom]]
                                  轉換成字串
       ];
       Return[returnedString]
              返回
      ];
    ]
   ]
  ]
```

Import data

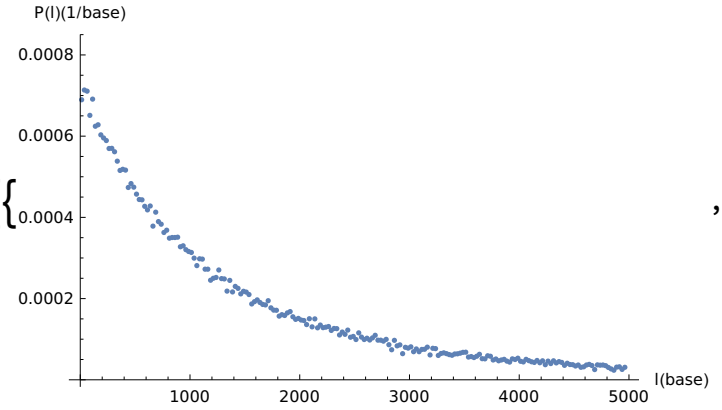
In[8]:=
```
recombinationRates = {1*^-7, 1*^-8, 1*^-9};
populationSize = 1*^4;
```

In[3]:=
```
histograms = Table[0, {idx, recombinationRates}];
                   表格
Do[histograms[[idx]] = Transpose[Interpreter[
Do迴圈                  轉置          解譯器
     DelimitedSequence[DelimitedSequence["Number", {"[", ", ", "]"}], {"[", ", ", "]"}]][
     分隔序列                 分隔序列                  數
   Import[StringJoin[NotebookDirectory[], number2Printed[recombinationRates[[idx]]],
   導入     字串結合        筆記本目錄
     "_histogram.txt"]]]], {idx, Length[recombinationRates]}]
                                  長度
```

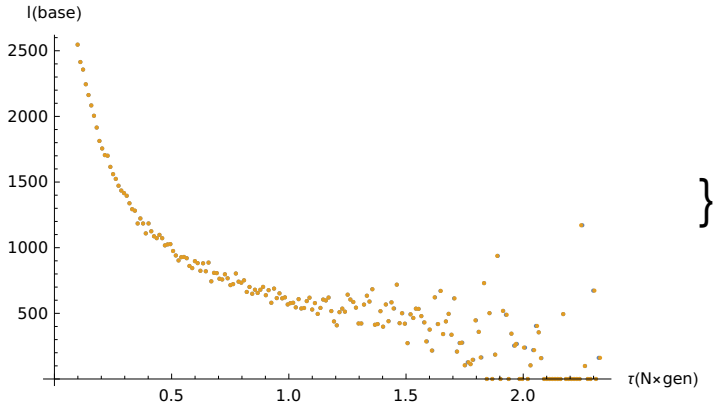
In[5]:=
```
data = Table[0, {idx, recombinationRates}];
             表格
Do[data[[idx]] = Interpreter[
Do迴圈              解譯器
     DelimitedSequence[DelimitedSequence["Number", {"[", ", ", "]"}], {"[", ", ", "]"}]][
     分隔序列                 分隔序列                  數
   Import[StringJoin[NotebookDirectory[], number2Printed[recombinationRates[[idx]]],
   導入     字串結合        筆記本目錄
     "_times.txt"]]], {idx, Length[recombinationRates]}]
                                  長度
```

In[19]:=
```
Table[
 {ListPlot[histograms[[idx]], ImageSize → Medium, AxesLabel → {"l(base)", "P(l)(1/base)"}],
  ListPlot[{Transpose[{data[[idx, 1]] / populationSize, data[[idx, 3]]}],
    Transpose[{data[[idx, 2]] / populationSize, data[[idx, 3]]}]},
   ImageSize → Medium, AxesLabel → {"τ(N×gen)", "l(base)"}]}, {idx, 3}]
```

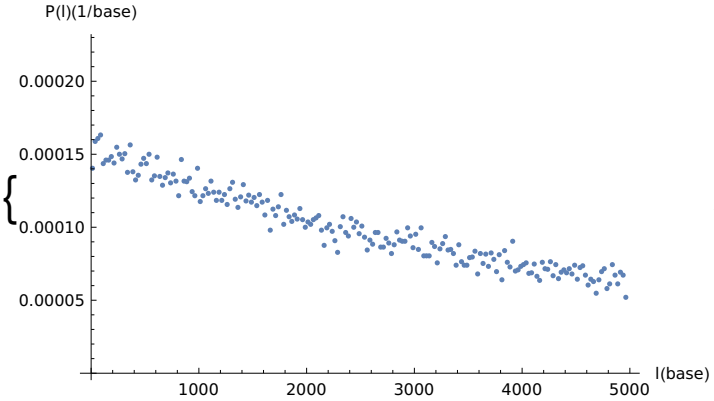
{{

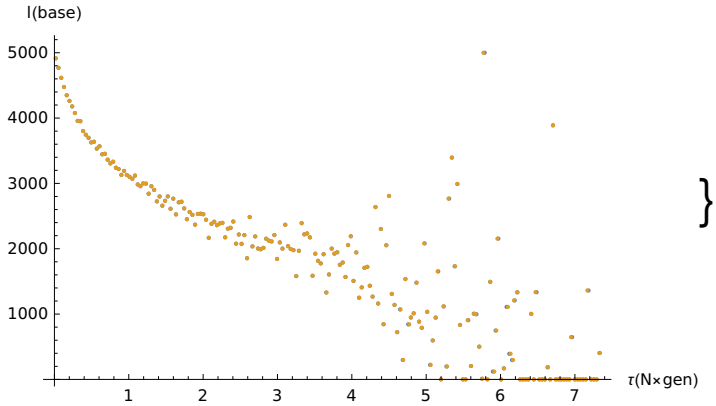


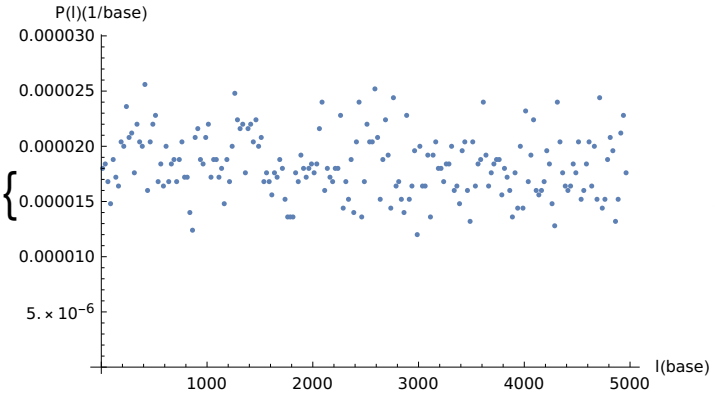
,

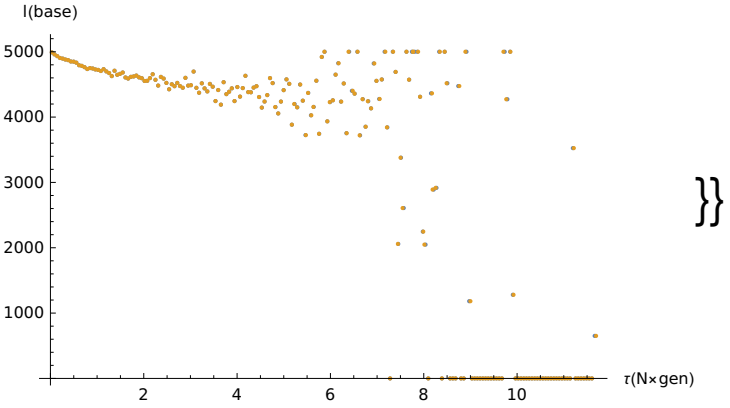


},

{ 
P(l)(1/base)
0.00020
0.00015
0.00010
0.00005
1000
2000
3000
4000
5000
l(base)
,


l(base)
5000
4000
3000
2000
1000
1
2
3
4
5
6
7
τ(N×gen)
},

{ 
P(l)(1/base)
0.000030
0.000025
0.000020
0.000015
0.000010
5. × 10−6
1000
2000
3000
4000
5000
l(base)
,


l(base)
5000
4000
3000
2000
1000
2
4
6
8
10
τ(N×gen)

}}