Genotype Likelihood Estimation

Andreas Füglistaler, PhD

Wegmann Group

UNI FR

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Calling Genotype

Read	Base
1	A
2	G
3	T
4	A
5	A
6	G
7	G
8	A
9	A
10	G

➡

Calling Genotype

Read	Base
1	A
2	G
3	T
4	A
5	A
6	G
7	G
8	A
9	A
10	G

➡ AG

Calling Genotype

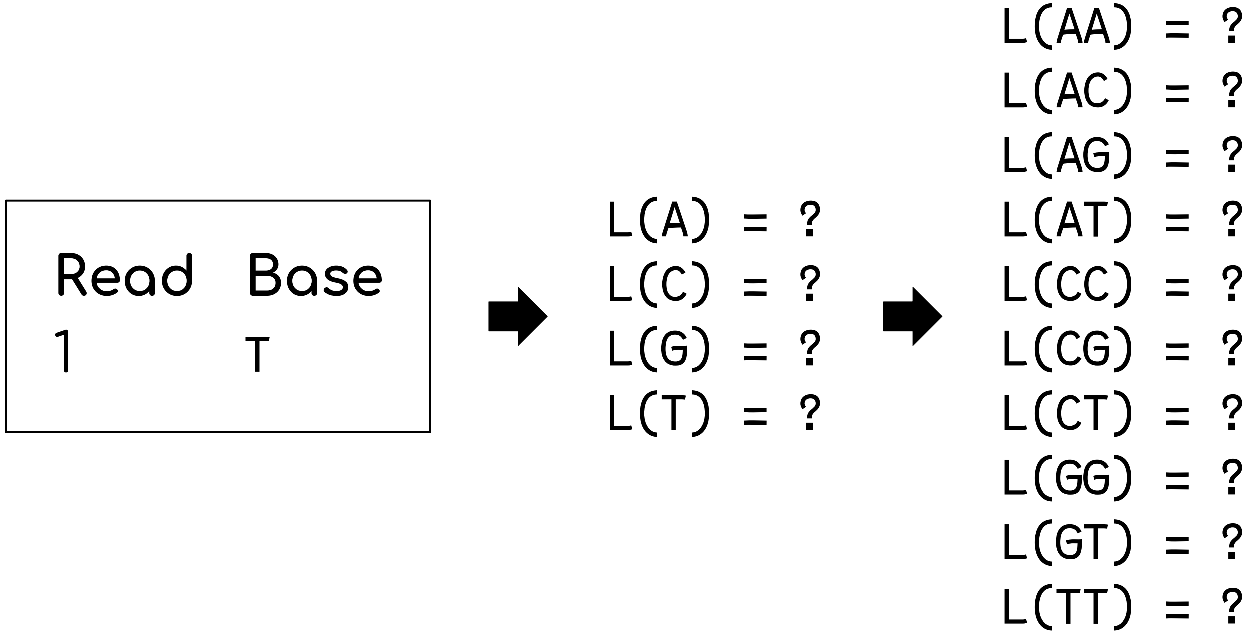
Read	Base
1	T

➡ AT, CT, GT, TT ?

Is it even a T?

Likelihoods

Base Likelihoods: `L(A), L(C), L(G), L(T)`
Genotype Likelihood: `L(ab) = 0.5×[L(a) + L(b)]`



Read	Base
1	T

```
L(A) = ?
L(C) = ?
L(G) = ?
L(T) = ?
```

```
L(AA) = ?
L(AC) = ?
L(AG) = ?
L(AT) = ?
L(CC) = ?
L(CG) = ?
L(CT) = ?
L(GG) = ?
L(GT) = ?
L(TT) = ?
```

Genotype Likelihoods

Assuming no Errors

Read	Base
1	T

➡

L(A) = 0
L(C) = 0
L(G) = 0
L(T) = 1

➡

L(AA) = 0
L(AC) = 0
L(AG) = 0
L(AT) = 0.5×(0 + 1)
L(CC) = 0
L(CG) = 0
L(CT) = 0.5×(0 + 1)
L(GG) = 0
L(GT) = 0.5×(0 + 1)
L(TT) = 1

Post-Mortem Damage

Deamination of Cytosine to Uracil: C→U

Uracil will be read as Thymine: C→U→T

Estimation of C→T transition

Position: Distance from 5' read end

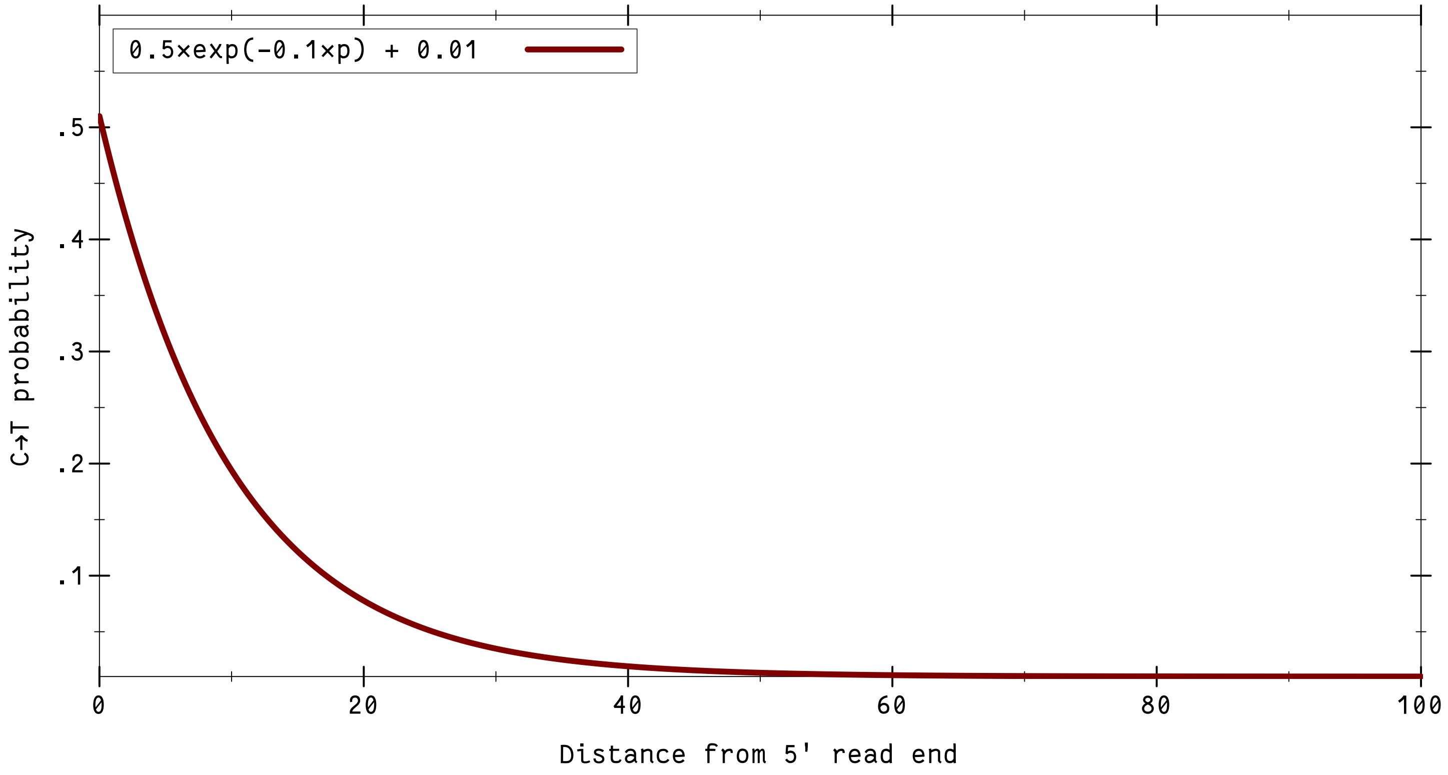
For every C in the reference, count occurrence in data

- Number of C→T per position
- Total number of Cs per position

```
PMD(C→T, p) = Number(C→T, p)/tot(C, p)
```

- Either empiric values or fit exponential function
- (Same for G→A from 3' if paired ended reads)

Post-Mortem Damage



Genotype Likelihoods with PMD

Assuming PMD(C→T) = 0.3

Read	Base
1	T

➡

```
L(A) = 0
L(C) = 0.3
L(G) = 0
L(T) = 1
```

➡

```
L(AA) = 0
L(AC) = 0.5×(0 + 0.3)
L(AG) = 0
L(AT) = 0.5×(0 + 1)
L(CC) = 0.3
L(CG) = 0.5×(0.3 + 0)
L(CT) = 0.5×(0.3 + 1)
L(GG) = 0
L(GT) = 0.5×(0 + 1)
L(TT) = 1
```

Sequencing Errors

Reported error probability by sequencing machine:

$Q = -10 \times \log(\varepsilon)$

- Not very accurate
- Needs recalibration

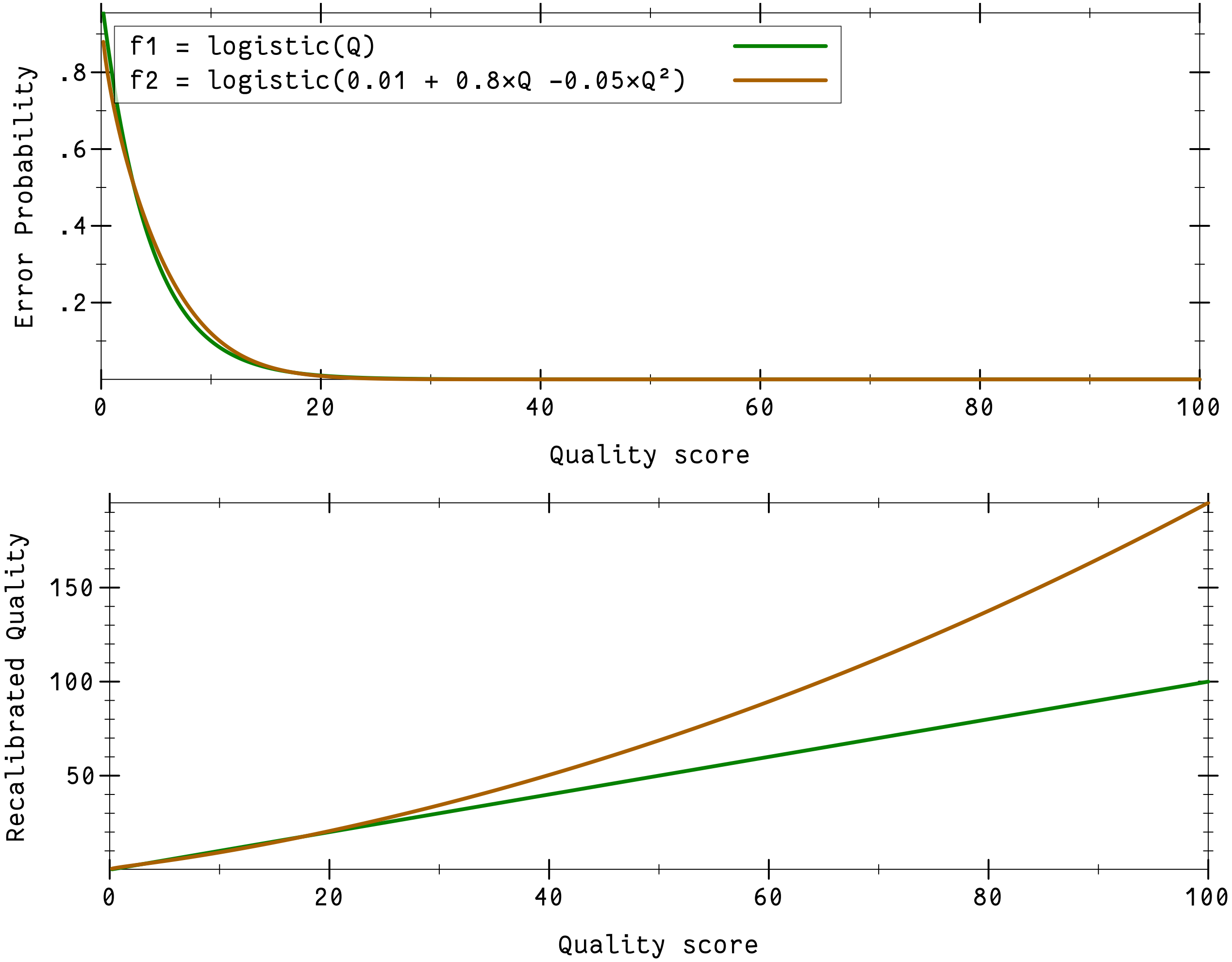
Estimate recalibration

- Use monomorphic/haploid sites

```
ε = logistic[f0 + f1(T(Q)) + f2(p) + f3(mappingQuality)
                + f4(fragmentLength) + f5(context)]
f = polynomial, empiric, probit or 0
ρ = [[-, A→C, A→G, A→T],
     [C→A, -, C→G, C→T],
     [G→A, G→C, -, G→T],
     [T→A, T→C, T→G, -]]
```

- Expectation–maximization (EM) algorithm

Sequencing Errors

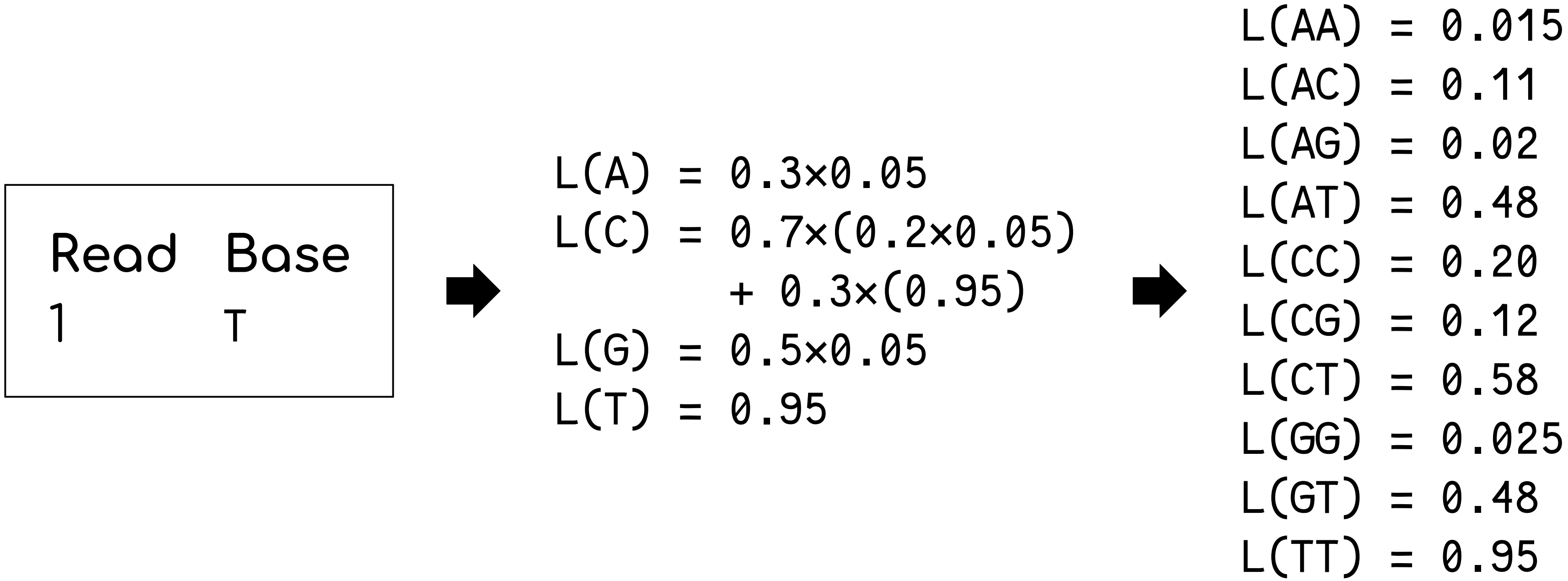


Genotype Likelihoods with Recal

Assuming:

PMD(C→T) = 0.3, ε = 0.05

ρ(A→T) = 0.3, ρ(C→T) = 0.2, ρ(G→T) = 0.5



Read	Base
1	T

```
L(A) = 0.3×0.05
L(C) = 0.7×(0.2×0.05)
          + 0.3×(0.95)
L(G) = 0.5×0.05
L(T) = 0.95
```

```
L(AA) = 0.015
L(AC) = 0.11
L(AG) = 0.02
L(AT) = 0.48
L(CC) = 0.20
L(CG) = 0.12
L(CT) = 0.58
L(GG) = 0.025
L(GT) = 0.48
L(TT) = 0.95
```

ATLAS

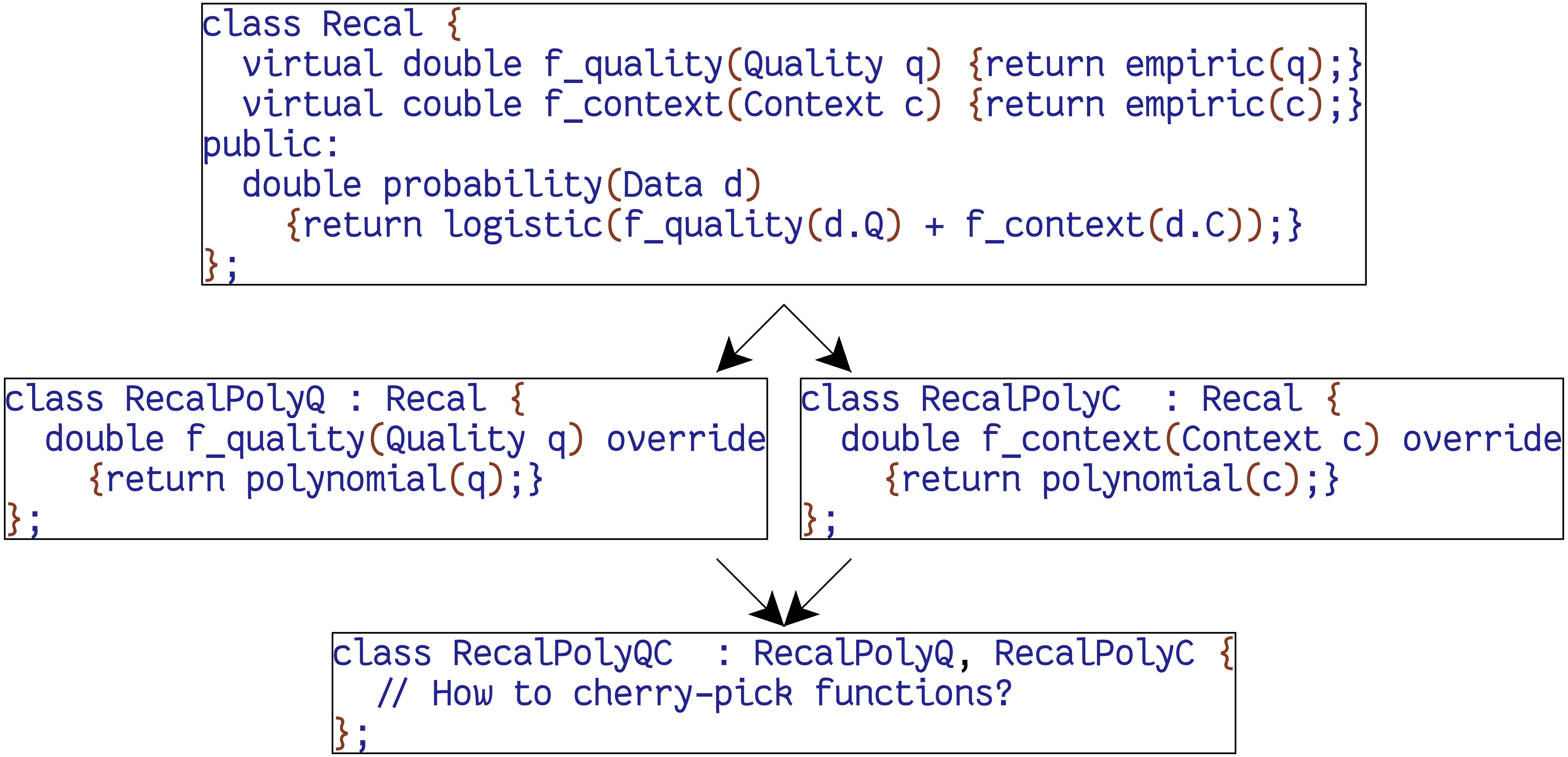
Analysis Tools for Low-coverage and Ancient Samples

48 Tasks

- call, theta, inbreeding, GLF, majorMinor, ...

- Simulate data
- Estimate PMD
- Estimate sequencing error recalibration

Implementation Inheritance



```
class Recal {
  virtual double f_quality(Quality q) {return empiric(q);}
  virtual couble f_context(Context c) {return empiric(c);}
public:
  double probability(Data d)
    {return logistic(f_quality(d.Q) + f_context(d.C));}
};
```

```
class RecalPolyQ : Recal {
  double f_quality(Quality q) override
    {return polynomial(q);}
};
```

```
class RecalPolyC  : Recal {
  double f_context(Context c) override
    {return polynomial(c);}
};
```

```
class RecalPolyQC  : RecalPolyQ, RecalPolyC {
  // How to cherry-pick functions?
};
```

Implementation Inheritance: Pro & Contra

✔ ’Natural evolution’ from mono- to polymorphic
✔ Straightforward to implement
✔ Works well in small, easy cases

x Multiplicative complexity (NxM implementations)
x Long inheritance chains
x Diamond inheritance problem
x Magohamoth-sized classes
x ’But I only want feature a, not a, b, c & d!’

Interface Inheritance

```
struct QualityFn {virtual double apply(Quality q) = 0;};
struct ContextFn {virtual double apply(Context c) = 0;};
```

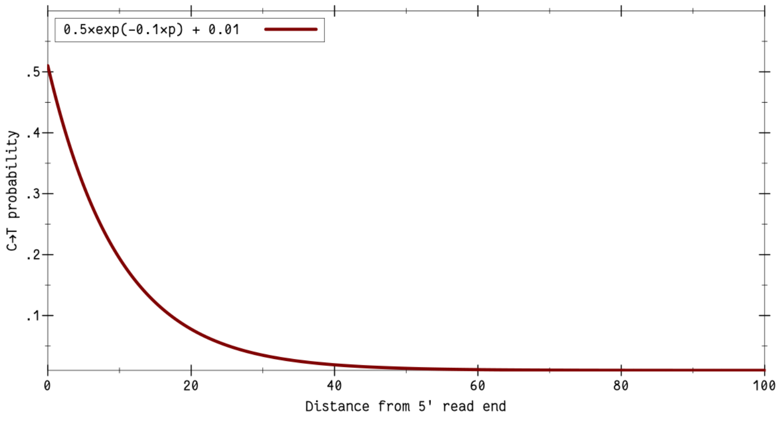
```
class Recal final {
  QualityFn* qf;
  ContextFn* cf;
public:
  Recal(QualityFn* q, ContextFn* c) {qf = q; cf = c;}
  double probability(Data d)
    {return logistic(qf->apply(d.Q) + cf->apply(d.C));}
};
```

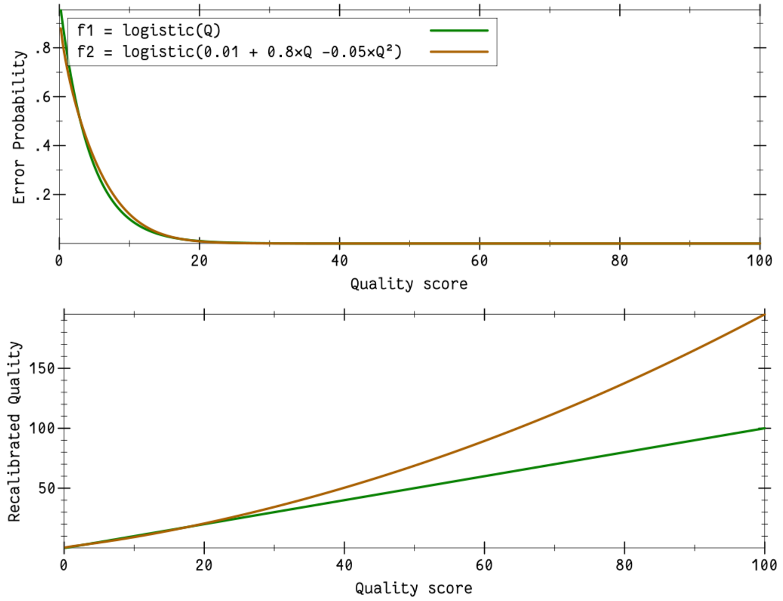
```
class EmpiricQuality final: QualityFn {
  double apply(Quality q) override
    {return empiric(q);}
};

class PolyQuality final : QualityFn {
  double apply(Quality q) override
    {return polynomial(q);}
};
```

```
class EmpiricContext final : ContextFn {
  double apply(Context c) override
    {return empiric(c);}
};

class PolyContext final : ContextFn {
  double apply(Context c) override
    {return polynomial(c);}
};
```

Simulation

```
~/Git/atlas/build/atlas --task simulate --ploidy 2,2,1 --depth 2 --chrLength 500000
  --pmd "doubleStrand:Exponential[50,0.5,0.1,0.01]:Exponential[50,0.5,0.1,0.01]"
  --recal "intercept[0.1];quality:polynomial[0.8,-0.05]"
```





Estimate PMD pattern

```
~/Git/atlas/build/atlas --task PMD --bam *.bam --fasta *.fasta
  --pmdModels "doubleStrand:Exponential:Exponential"
```

also possible

```
  --pmdModels "singleStrand:Empiric:Empiric"
```

Estimate recalibration Pattern

```
~/Git/atlas/build/atlas --task recal --bam *.bam  --regions chr3.bed
  --pmd *_PMD.txt --recal "intercept;quality:polynomial2"
```

also possible

```
 --recal "intercept;quality:empiric"
 --recal "intercept;quality;position;context;fragmentLength;mappingQuality"
 --recal "intercept;quality:polynomial3;fragmentLength:probit;context"
```

Estimate θ

```
~/Git/atlas/build/atlas --task theta --bam *.bam
```

```
~/Git/atlas/build/atlas --task theta --bam *.bam --pmd *_PMD.txt
```

```
~/Git/atlas/build/atlas --task theta --bam *.bam
  --pmd *_PMD.txt --recal *_recal.txt
```

Calculating Genotype Likelihoods

1. Estimate PMD pattern

Covariate: Position

- PMD(C→T) = Number(C→T)/Number(C)

2. Estimate Sequencing Error recalibration

Covariates: Sequencing quality, Mapping quality,
Context, Position, Fragment length

- Use monomorphic/haploid sites
- EM on multi-variate recalibration function

3. Estimate Genotype Likelihoods

- θ, inbreeding coefficient, ...