The biochemical resolving power of fluorescence lifetime imaging

Andrew *L.* Trinh[1,*] and Alessandro Esposito[1]
[1]MRC Cancer Unit, University of Cambridge, Cambridge, United Kingdom
*at862@cam.ac.uk

Keywords: fluorescence lifetime, resolution, microscopy, biochemistry

Published in bioRxiv: doi__________

In[•]:=
```
(* INITIALIZATION *)
(* Define subcripted symbols *)
<< Notation`
Symbolize[ σ_ ]
(* Beep at every evaluation *)
$Post = (Beep[]; #) &;
```

In[•]:=
```
(* Define positivity of all variables, and set substitution rules *)
opts = {B > 0, m > 1, i > 1, τ > 0, A > 0,
   T > 0, C > 0, t ≥ 0, p > 0, σirf > 0, ρ > 0, ω > 0, μ > 0, η > 0};
```

$$\texttt{subs} = \left\{\tau \to \frac{\sigma_{irf}}{\sqrt{2}\,\rho},\ T \to \frac{\omega\sqrt{2}\,\sigma_{irf}}{m},\ A \to 2\,\frac{B}{\tau},\ \mu \to \eta\sqrt{2}\,\sigma_{irf}\right\};$$

$$\texttt{subsinv} = \left\{\rho \to \frac{\sigma_{irf}}{\sqrt{2}\,\tau},\ \omega \texttt{->} \frac{m\,T}{\sqrt{2}\,\sigma_{irf}},\ B \to \frac{A\,\tau}{2},\ \eta \to \frac{\mu}{\sqrt{2}\,\sigma_{irf}}\right\};$$

In[•]:=
```
(* Exponential decay with Gaussian IRF *)
f1 = FullSimplify[Convolve[PDF[NormalDistribution[μ, σirf], x],
   UnitStep[x] A Exp[-x / τ], x, y, Assumptions → opts], Assumptions → opts]
```

Out[•]=

$$\frac{1}{2} A\, e^{\frac{\sigma_{irf}^2 - 2 y \tau + 2 \mu \tau}{2 \tau^2}}\, \mathrm{Erfc}\left[\frac{\sigma_{irf}^2 + (-y + \mu)\,\tau}{\sqrt{2}\,\sigma_{irf}\,\tau}\right]$$

In[•]:=
```
(* Expected counts within interval t, starting from time 0 *)
Λ[t_] =
 Simplify[Integrate[f1, {y, 0, t}, Assumptions → opts] //. subs, Assumptions → opts]
```

Out[•]=

$$B\left(\mathrm{Erf}[\eta] - \mathrm{Erf}\left[\eta - \frac{t}{\sqrt{2}\,\sigma_{irf}}\right] + e^{\rho\,(2\eta+\rho)}\,\mathrm{Erfc}[\eta+\rho] - e^{\rho\left(2\eta+\rho-\frac{\sqrt{2}\,t}{\sigma_{irf}}\right)}\,\mathrm{Erfc}\left[\eta+\rho-\frac{t}{\sqrt{2}\,\sigma_{irf}}\right]\right)$$

In[∘]:=
```
(* Log-likelihood. Terms that do not depend on B or ρ are neglected *)

(* The derivative of the logarithm generate a denumerator that causes nbumerical
 instabilities. REGCNST avoids indeterminate cases of the type 1/0*)
REGCNST = 1*^-6;

LogL[B_, ρ_] =
 Simplify[(∑_{i=1}^{m} Log[REGCNST + Λ[i T] - Λ[(i - 1) T]] G[i] - ∑_{i=1}^{m} (Λ[i T] - Λ[(i - 1) T])) //.
   T → ω √2 σ_irf / m //. η → 0, Assumptions → opts]
```

Out[∘]=

$$-B\left(\text{Erf}[\omega]+e^{\rho^2}\,\text{Erfc}[\rho]-e^{\rho\,(\rho-2\,\omega)}\,\text{Erfc}[\rho-\omega]\right)+\sum_{i=1}^{m}G[i]\,\text{Log}\Big[\frac{1}{1\,000\,000}-B\left(\text{Erf}\Big[\frac{(-1+i)\,\omega}{m}\Big]+e^{\rho^2}\,\text{Erfc}[\rho]-e^{\rho\left(\rho-\frac{2\,(-1+i)\,\omega}{m}\right)}\,\text{Erfc}\Big[\rho-\frac{(-1+i)\,\omega}{m}\Big]\right)+B\left(\text{Erf}\Big[\frac{i\,\omega}{m}\Big]+e^{\rho^2}\,\text{Erfc}[\rho]-e^{\rho\left(\rho-\frac{2\,i\,\omega}{m}\right)}\,\text{Erfc}\Big[\rho-\frac{i\,\omega}{m}\Big]\right)\Big]$$

In[∘]:=
```
(* Defines the counts per bin, keeping this implicit to avoid being
 differentiated. EG will be replaced to G[i] when computing expectations *)

EGi = FullSimplify[(Λ[i T] - Λ[(i - 1) T]) //. subs //. η → 0, Assumptions → opts]

(* η→0 is the particular case where the is no offset on the temporal axis *)
```

Out[∘]=

$$-B\left(-1+\text{Erf}\Big[\frac{(-1+i)\,\omega}{m}\Big]+\text{Erfc}\Big[\frac{i\,\omega}{m}\Big]+e^{\rho\left(\rho-\frac{2\,i\,\omega}{m}\right)}\left(\text{Erfc}\Big[\rho-\frac{i\,\omega}{m}\Big]-e^{\frac{2\,\rho\,\omega}{m}}\,\text{Erfc}\Big[\rho+\frac{\omega-i\,\omega}{m}\Big]\right)\right)$$

In[∘]:=
```
(* Evaluate derivatives, elements of the Fisher information matrix *)

a = -D[LogL[B, ρ], {ρ, 2}];
b = -D[LogL[B, ρ], {B, 2}];
c = -D[LogL[B, ρ], {B, 1}, {ρ, 1}];
```

In[ ]:=
```
(*F is F2→C στ^2/τ^2;*)

(* Error propagation *)
(* στ^2=((D[τ/.subs[[1]],{ρ,1}])^2/.subsinv)σρ^2->(2 τ^4/σirf^2)σρ^2;*)
(* Fisher Information matrix inversion, firt element*)
(* σρ=Sqrt[b/(a b - c^2)/.G[i]→EGi/.η→0]//.subsinv; *)

(* Expectations will be implicetly determined
 using Gi instead of G[i] and substituting B for  *)

EB = Solve[C == Λ[m T] //. subs /. η → 0, B][[1]][[1]]

(* Therefore, evaluating expectations by substitution with EG and EB *)
F2ev = C (2 τ^2/σirf^2) (b/(a b - c^2)) /. {G[i] → EGi} /. EB /. {ρ → σirf/(√2 τ), ω -> PER/(√2 σirf)};
```

Out[ ]= $B \to \dfrac{C}{\mathrm{Erf}[\omega] + e^{\rho^2}\,\mathrm{Erfc}[\rho] - e^{\rho\,(\rho-2\,\omega)}\,\mathrm{Erfc}[\rho-\omega]}$

In[ ]:=
```
(* Test *)
(* Approx solutions show that F2ev does not depend
 on C. This line is used only for checking  that F2ev is well
 behaved numerically and confirming the independency from C *)

F2ev /. {σirf → .1, τ → .01, PER → 25, m → 128, C → 10}
```

General: Exp[-2450.] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-2450.] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-2450.] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

Out[ ]= 400.138

```
(* Control curves (Kollner&Wolfrum *)
F2Dirac = ExpandAll[
```

$$\left(\left(-1+e^{T/\tau}\right)^2\left(-1+e^{\frac{m T}{\tau}}\right)^2\tau^2\right)\Big/\left(\left(e^{T/\tau}+e^{\frac{T+2 m T}{\tau}}-e^{\frac{m T}{\tau}}m^2-e^{\frac{(2+m) T}{\tau}}m^2+2\,e^{\frac{(1+m) T}{\tau}}\left(-1+m^2\right)\right)T^2\right)];$$

In[•]:=
```
τn = 151;
mn = 12;
σn = 7;

memo = Table[100000, τn, mn, σn];
memo2 = Table[100000, τn, mn, σn];



σ0 = {0.001, 0.1, 0.25, 0.5, 1, 2, 5};
τ0 = Table[0.01 * 1.05^(i - 1), {i, τn}];
m0 = Table[2^i, {i, mn}];

For[τi = 1, τi ≤ τn, τi++,
 For[mi = 1, mi ≤ mn, mi++,
   For[σi = 1, σi ≤ σn, σi++,
    vals = {C → 1, σirf → σ0[[σi]], τ → τ0[[τi]], PER → 25, m → m0[[mi]]};
    tmp = (F2ev /. vals);
    memo2[[τi, mi, σi]] = F2Dirac /. {T → PER / m} //. vals;
    If[tmp > 0.001, memo[[τi, mi, σi]] = tmp]
   ]
  ]
  Print[τi]
]
Beep[]
```

General: Exp[−1250.] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−1250.] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−2500.] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

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

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

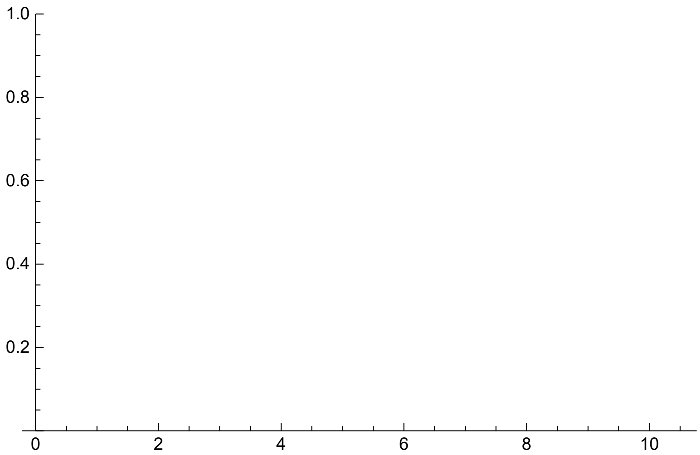
151

In[∘]:=
```
Export["GaussianForMatlab.mat", memo]
```
Out[∘]=
```
GaussianForMatlab.mat
```

```
Export["GaussianForMathematica.dat", All]
```

In[∘]:=
```
memo[[1, 1, 1]]
```
Out[∘]=
```
4072.91
```

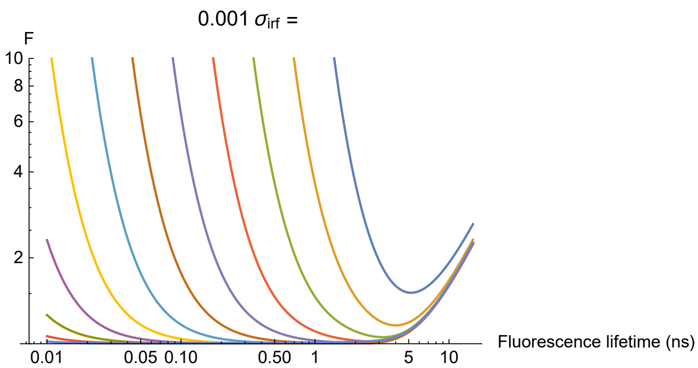
In[∘]:=
```
ListPlot[Table[{τ0[[τi]], memo[[τi, mi, 1]]}, {mi, 1, mn}, {τi, 1, τn}],
 Joined → True, PlotStyle -> Directive[Black, Thin, SolidData], PlotRange → {0, 1}]
```
Out[∘]=



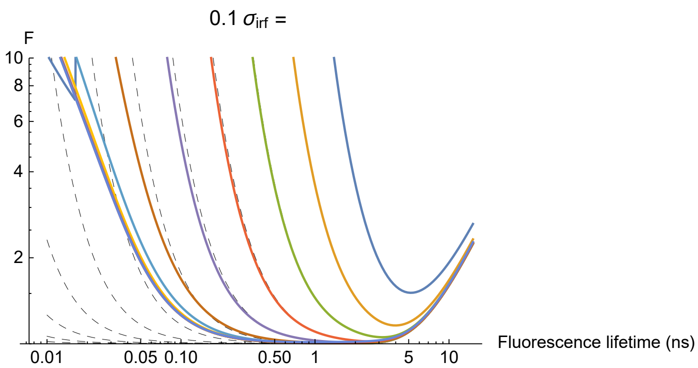
In[ ]:=

```
pr = ListLogLogPlot[
   Table[{τ0[[τi]], √memo[[τi, mi, 1]] }, {mi, 1, mn}, {τi, 1, τn}], PlotRange → {1, 10},
   Joined → True, PlotStyle -> Directive[Black, Thickness[.001], Dashed]];
For[j = 1, j ≤ σn, j++, p0 = ListLogLogPlot[Table[{τ0[[τi]], 1}, {mi, 1, 1}, {τi, 1, τn}],
     PlotRange → {1, 10}, Joined → True, PlotStyle -> Directive[Black, Thin, SolidData]];
  p1 = ListLogLogPlot[Table[{τ0[[τi]], √memo[[τi, mi, j]] }, {mi, 1, mn}, {τi, 1, τn}],
     PlotRange → {1, 10}, Joined → True,
     PlotStyle -> Directive[Thickness[.005], SolidData]];
  p2 = ListLogLogPlot[Table[{τ0[[τi]], √memo2[[τi, mi, j]] }, {mi, 1, mn}, {τi, 1, τn}],
     Joined → True, PlotStyle -> Directive[Thin, Dashed]];
  Print[Show[pr, p1, PlotLabel → "σirf = " σ0[[j]],
     AxesLabel → {"Fluorescence lifetime (ns)", "F"}]];
  Print[25. / m0]

 ];
```

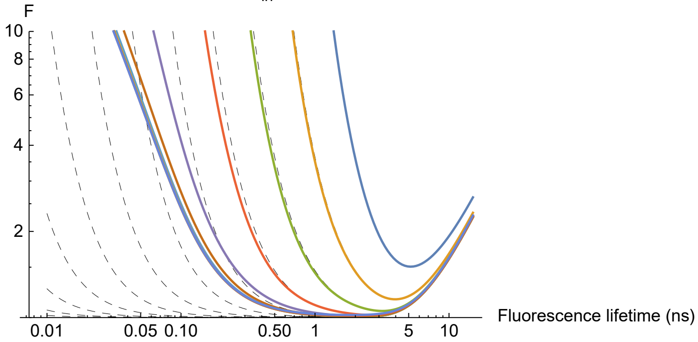


{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625,
 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}



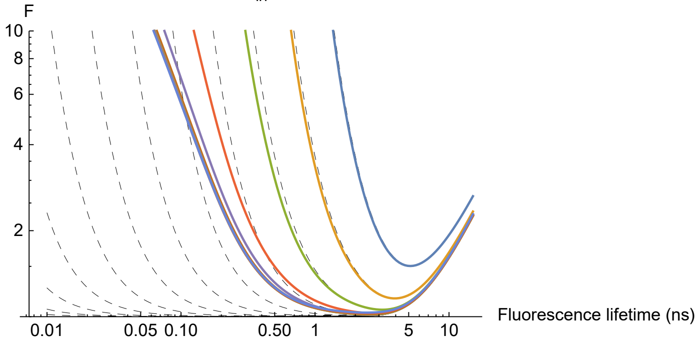
{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625,
 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}

$0.25\,\sigma_{irf}$ =



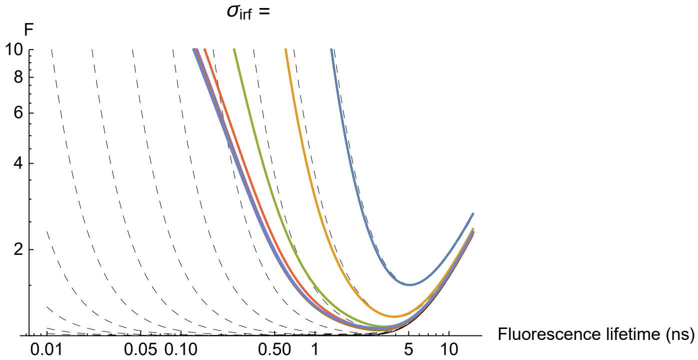
{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}

$0.5\,\sigma_{irf}$ =



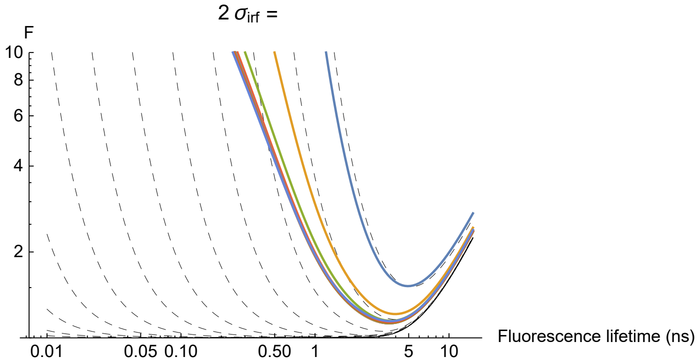
{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}

σ_{irf} =

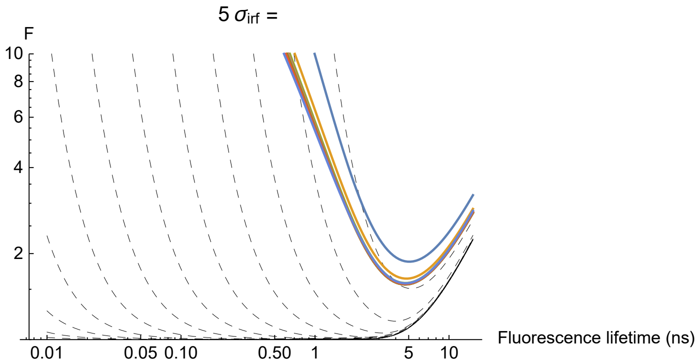


{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}

$2\,\sigma_{irf}$ =



{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}




```
{12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625,
 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}
```

In[ ]:=

In[ ]:= **25. / m0**

```
Out[ ]= {12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625,
 0.195313, 0.0976563, 0.0488281, 0.0244141, 0.012207, 0.00610352}
```

(* IRF has to be sampled properly. All curves with PER/m *)

In[ ]:= **NotebookSave[]**