

Functions

ln[*]:=

```

H[θ_] := H[θ] =  $\frac{\theta}{1 + \theta}$ 

Ho[θ_, ρ_] :=
  Ho[θ, ρ] =  $\frac{18 + 13 \rho + \rho^2 + 36 \theta + 22 \theta^2 + 4 \theta^3 + \rho (6 \theta + \theta^2)}{(1 + \theta) (18 + 13 \rho + \rho^2 + 54 \theta + 40 \theta^2 + 8 \theta^3 + \rho (\rho \theta + 19 \theta + 6 \theta^2))}$ 

H2[θ_, ρ_] := H2[θ, ρ] =
   $\frac{\theta^2 (36 + 14 \rho + \rho^2 + 36 \theta + 6 \rho \theta + 8 \theta^2)}{(1 + \theta) (18 + 13 \rho + \rho^2 + 54 \theta + 40 \theta^2 + 8 \theta^3 + \rho (\rho \theta + 19 \theta + 6 \theta^2))}$ 

H1[θ_, ρ_] := H1[θ, ρ] = 1 - Ho[θ, ρ] - H2[θ, ρ]

re[d_, g_, rbp_, L_] := rbp  $\left( d + 2 \left( \frac{g}{rbp} \right) L \left( 1 - e^{\left( \frac{-d}{L} \right)} \right) \right)$ 

H[θ_] :=  $\frac{\theta}{1 + \theta}$ 

Ho[θ_, d_, g_, rbp_, L_] :=
  (18 + 13 re[d, g, rbp, L] + re[d, g, rbp, L]^2 + 36 θ + 22 θ^2 +
    4 θ^3 + re[d, g, rbp, L] (6 θ + θ^2)) /
  ((1 + θ) (18 + 13 re[d, g, rbp, L] + re[d, g, rbp, L]^2 + 54 θ + 40 θ^2 +
    8 θ^3 + re[d, g, rbp, L] (re[d, g, rbp, L] θ + 19 θ + 6 θ^2)))

H2[θ_, d_, g_, rbp_, L_] :=
  (θ^2 (36 + 14 re[d, g, rbp, L] + re[d, g, rbp, L]^2 + 36 θ + 6 re[d, g, rbp, L] θ +
    8 θ^2)) / ((1 + θ) (18 + 13 re[d, g, rbp, L] + re[d, g, rbp, L]^2 + 54 θ +
    40 θ^2 + 8 θ^3 + re[d, g, rbp, L] (re[d, g, rbp, L] θ + 19 θ + 6 θ^2)))

H1[θ_, d_, g_, rbp_, L_] := 1 - Ho[θ, re[d, g, rbp, L]] - H2[θ, re[d, g, rbp, L]]

```

(*Expectations replacing rho with re*)

ln[*]:=

```

(*note pooling individuals for a single individual just formats the data
  from the 'dataset' for the functions below calculating likelihoods*)
PoolIndividuals[chr_] := Module[{chrUnbinned},
  chrUnbinned = chr[[All, 3 ;;]];
  chrUnbinned = {#[[1]][[1]], Total[#[[All, 2]]], Total[#[[All, 3]]], Total[#[[All, 4]]],

```

```

      Mean[#[[All, -1]]] & /@ (Values @ GroupBy[chrUnbinned [[All, All]], First]);
chrUnbinned
];
BinnedChrCounts [chrUnbinned_ , binSize_] :=
Module[{interval = binSize, binned1, binned2},
  binned1 = Table[{#[[1]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[2]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[3]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[4]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    {i, 1, Floor @ (Length[chrUnbinned] / interval)}}];
  binned1 = SortBy[binned1, First];
  binned2 = {Mean[#[[1]]], Total[#[[2]]], Total[#[[3]]], Total[#[[4]]]} & /@ binned1;
  binned2
]
BinnedChrProp [chrUnbinned_ , binSize_] :=
Module[{interval = binSize, binned1, binned2},
  binned1 = Table[{#[[1]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[2]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[3]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    #[[4]] & /@ chrUnbinned [[interval (i - 1) + 1 ;; interval (i)]],
    {i, 1, Floor @ (Length[chrUnbinned] / interval)}}];
  binned1 = SortBy[binned1, First];
  binned2 = {Mean[#[[1]]], Total[#[[2]]], Total[#[[3]]], Total[#[[4]]]} & /@ binned1;
  binned2 = {#[[1]], #[[2]] / Total[#[[2 ;; 4]]],
    #[[3]] / Total[#[[2 ;; 4]]], #[[4]] / Total[#[[2 ;; 4]]]} & /@ binned2;
  binned2
]
GetChrLikeFun [binnedChr_] := Module[{binned2 = binnedChr , binnedLike},
  binnedLike = Sum[ binned2 [[i, 2]] * Log[ Ho[ $\theta$ , binned2 [[i, 1]], g, rbp, L]] +
    binned2 [[i, 3]] Log[H1[ $\theta$ , binned2 [[i, 1]], g, rbp, L]] +
    binned2 [[i, 4]] Log[H2[ $\theta$ , binned2 [[i, 1]], g, rbp, L]], {i, 1, Length[binned2]};
  binnedLike
];
CalculateLikeCounts [unbinnedChr_ , binSize_] := Module[
  {binnedCounts = GetChrLikeFun [BinnedChrCounts [unbinnedChr , binSize]],
  countEstimate , , propEstimate
},
  countEstimate = NMaximize[{binnedCounts
    , 0.0001 <  $\theta$  < 0.1, 25 ≤ L ≤ 500, 0.0000000001 < rbp < 0.1 , 0.0000001 < g < 0.1}
    { $\theta$ , rbp, g, L}, MaxIterations → 2000, PrecisionGoal → 10];
  countEstimate
];
CalculateLikeCountsAndPropsBinned [unbinnedChr_ , binSize_] := Module[

```

```

{binnedCounts = GetChrLikeFun [BinnedChrCounts [unbinnedChr , binSize]],
  binnedProp = GetChrLikeFun [BinnedChrProp [unbinnedChr , binSize]],
  countEstimate , , propEstimate
},
countEstimate = NMaximize[{binnedCounts
  , 0.0001 <  $\theta$  < 0.1, 25 ≤ L ≤ 500, 0.0000000001 < rbp < 0.1 , 0.0000001 < g < 0.1}
  { $\theta$ , rbp, g, L}, MaxIterations → 2000, PrecisionGoal → 10];
propEstimate = NMaximize[{binnedProp
  , 0.0001 <  $\theta$  < 0.1, 25 ≤ L ≤ 500, 0.0000000001 < rbp < 0.1 , 0.0000001 < g < 0.1}
  { $\theta$ , rbp, g, L}, MaxIterations → 2000, PrecisionGoal → 10];
{countEstimate , propEstimate}
];

```

Data:

```

In[ ]:= PoolIndividuals [chr_] := Module[{chrUnbinned},
  chrUnbinned = chr[[All, 3 ;;]];
  chrUnbinned = {#[[1]][[1]], Total[#[[All, 2]]], Total[#[[All, 3]]], Total[#[[All, 4]]],
    Mean[#[[All, -1]]]} & /@ (Values @ GroupBy[chrUnbinned [[All, All]], First]);
  chrUnbinned
];

GetChrLikeFun [binnedChr_] := Module[{binned2 = binnedChr , binnedLike},
  binnedLike = Sum[ binned2 [[i, 2]] * Log[ Ho[binned2[[i, 5]], binned2 [[i, 1]], g, rbp, L] ] +
    binned2 [[i, 3]] Log[H1[binned2[[i, 5]], binned2 [[i, 1]], g, rbp, L] + binned2 [[i, 4]]
    Log[H2[binned2[[i, 5]], binned2 [[i, 1]], g, rbp, L]], {i, 1, Length[binned2]}];
  binnedLike
];

```

```

In[ ]:= data =
  Import["~/Dropbox/projects/gene_conversion/mathematicaInputMice_unbinned .tsv",
    "Dataset", "HeaderLines" → 1];
fullDataCountsByDistance = PoolIndividuals [data[Select[#[chrom ≠ "chrX" &]]];

```

```

In[ ]:= estimatesChr19MinDist100Python =
  {0.002673853733400405 , 0.004444332823674365 , 113.24025414204982 }
  (* kappa, gamma, L*)(*here, distance is actually 101,
  not 100, but should make little difference*)

```

```

Out[ ]:= {0.00267385 , 0.00444433 , 113.24}

```

```
lnf := chr19 = PoolIndividuals [data[Select[#chrom == "chr19" &]]];
chr19MinDist100 = chr19[[100 ;; All]];

lnf := likeFunChr19 = GetChrLikeFun [chr19MinDist100 ]
```

239 345 024

$$\text{Log}\left[\left(0.992294 \left(18.2809 + 13.0467 \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right) rbp + \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right)^2 rbp^2\right) \right) / \right. \\ \left. \left(18.4218 + 13 \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right) rbp + \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right)^2 rbp^2 + \right. \right. \\ \left. \left. \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right) rbp \left(0.147906 + 0.00776547 \left(1000 + \frac{2(1-e^{-1000/L})gL}{rbp}\right) rbp\right)\right)\right] +$$

$$18.848 \text{ Log}\left[\frac{0.0000598379 \left(\dots 1 \dots\right)}{18.4218 + \dots 1 \dots + \dots 1 \dots + \dots 1 \dots}\right] + \dots 2699 \dots +$$

$$4.233.508 \text{ Log}\left[\dots 1 \dots\right] +$$

$$4.234.223 \text{ Log}\left[1 -$$

$$\frac{0.992343 \left(18.2791 + 13.0464 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right)^2 rbp^2\right)}{18.4191 + 13 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right)^2 rbp^2 + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp \left(0.146967 + 0.0077163 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp\right)} - \\ \frac{0.0000590853 \left(36.2783 + 14.0463 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right)^2 rbp^2\right)}{18.4191 + 13 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right)^2 rbp^2 + \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp \left(0.146967 + 0.0077163 \left(100 + \frac{2(1-e^{-100/L})gL}{rbp}\right) rbp\right)}$$

large output

[show less](#)

[show more](#)

[show all](#)

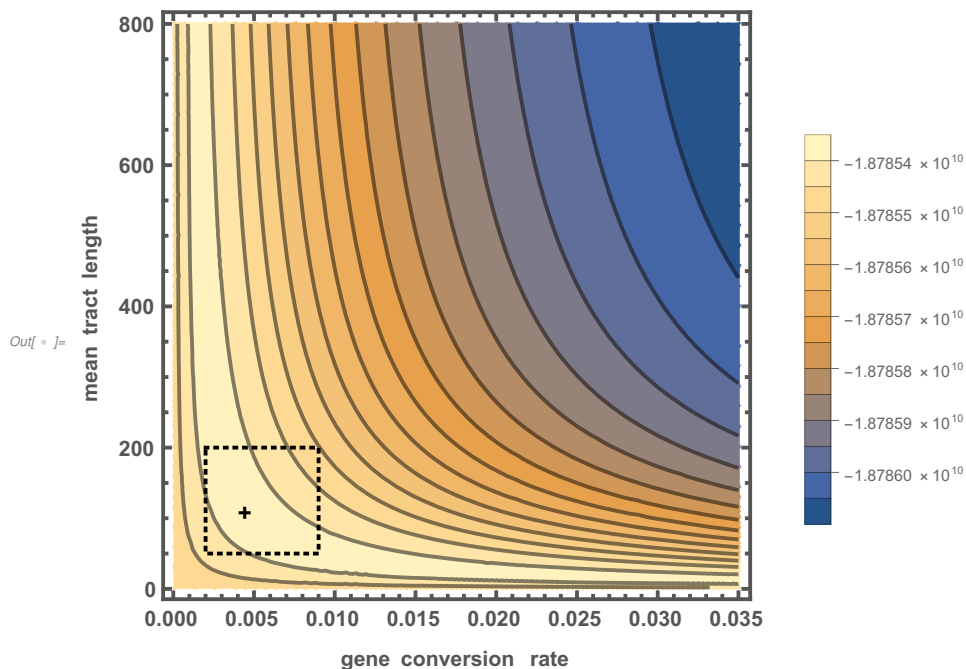
[set size limit...](#)

gc by L like surface

```

In[ ]:= pGcByLBig = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {rbp → 0.002673853733400405 }],
    {g, 0.0001, .035}, {L, 2, 800},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 20},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"gene conversion rate", "mean tract length"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog → {
      Inset[Graphics[{Black, Text[Style["+", 14, Bold]]}],
        {0.004444332823674365, 113.24025414204982 }]]
    ],
  Graphics[
    {EdgeForm[{Thick, Dotted, Black}], FaceForm[], Rectangle[{0.002, 50}, {0.009, 200}]}]
]

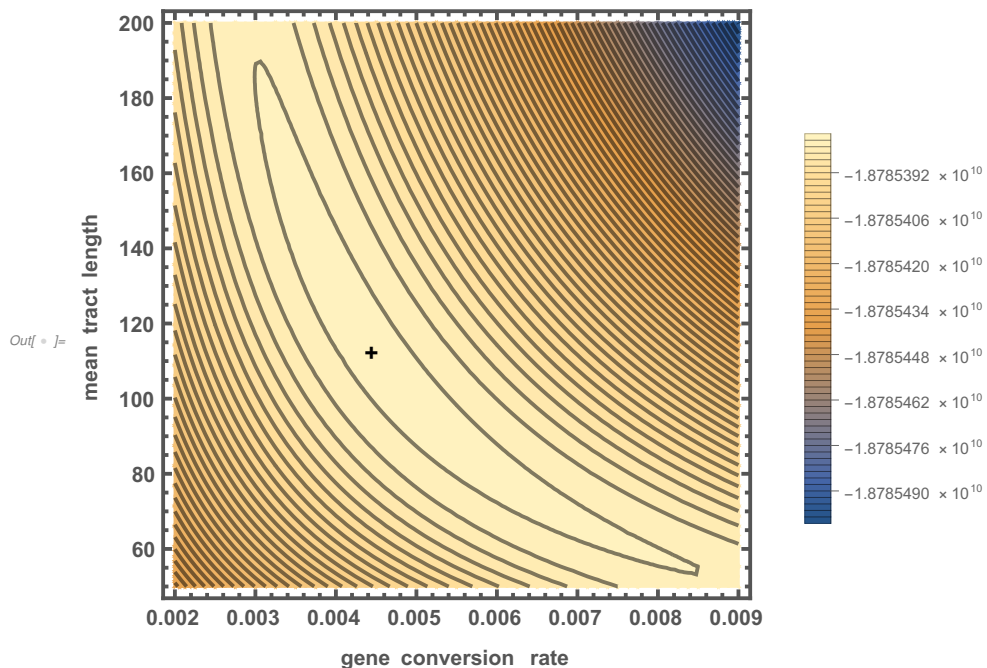
```



```

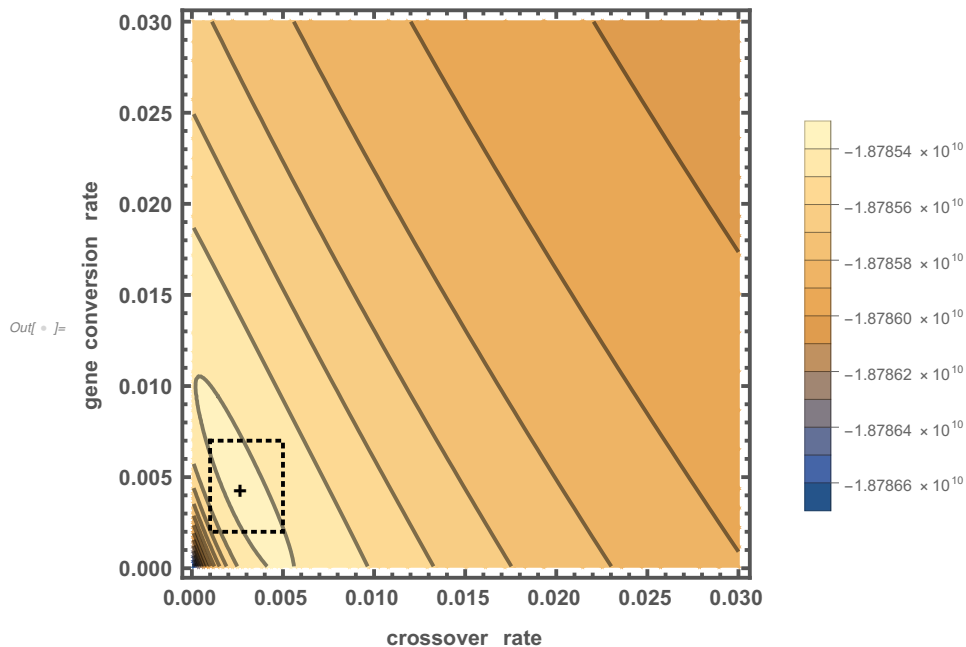
pGCByLSmall = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {rbp → 0.002673853733400405 }],
    {g, 0.002, 0.009}, {L, 50, 200},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 60},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"gene conversion rate", "mean tract length"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog → {
      Inset[Graphics[{Black, Text[Style["+" , 14, Bold]]}],
        {0.004444332823674365 , 113.24025414204982 }]
    ]
  ],
  Graphics[
    {EdgeForm[{Thick, Dotted, Black}], FaceForm[], Rectangle[{0.0018, 140}, {0.003, 280}]}]
]

```



XO by GC like surface

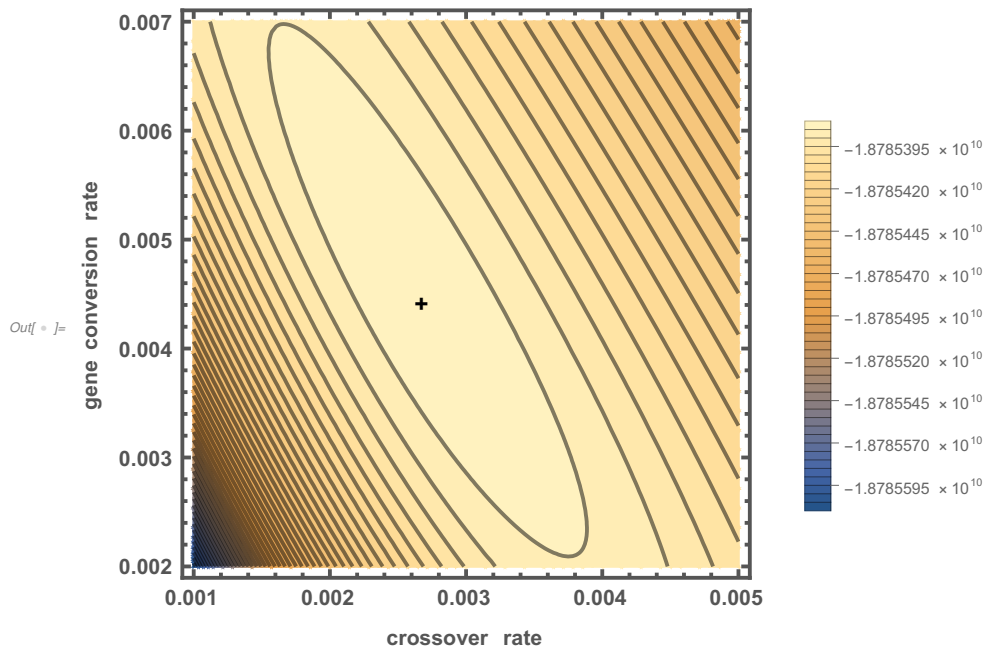
```
pXoByGcBig = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {L → 113.24}],
    {rbp, 0.0001, 0.03}, {g, 0.0001, 0.03},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 20},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"crossover rate", "gene conversion rate"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog →
      Inset[Graphics[{Black, Text[Style["+", 14, Bold]]}, {0.00267385, 0.00444433}]]],
  Graphics[{EdgeForm[{Thick, Dotted, Black}],
    FaceForm[], Rectangle[{0.001, 0.002}, {0.005, 0.007}]}]]
]
```



```

pXoByGcSmall = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {L → 113.24}],
    {rbp, 0.001, 0.005}, {g, 0.002, 0.007},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 60},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"crossover rate", "gene conversion rate"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog →
      Inset[Graphics[{Black, Text[Style["+", 14, Bold]]}], {0.00267385, 0.00444433}]
  ]
]

```

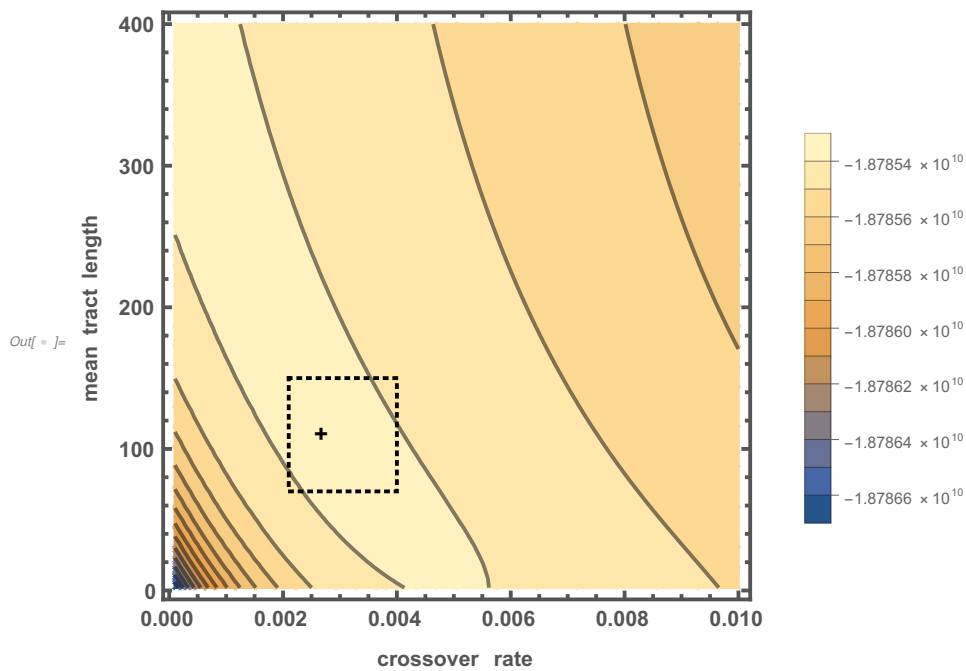


xo rate vs tract length

```

pXoByLBig = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {g → 0.00444433}],
    {rbp, 0.0001, .01}, {L, 2, 400},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 20},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"crossover rate", "mean tract length"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog → {
      Inset[Graphics[{Black, Text[Style["+", 14, Bold]]}],
        {0.002673853733400405, 113.24025414204982}],
    ],
  Graphics[
    {EdgeForm[{Thick, Dotted, Black}], FaceForm[], Rectangle[{0.0021, 70}, {0.004, 150}]}]
]

```



```

pXoByLSmall = Show[
  ContourPlot[
    Evaluate[likeFunChr19 /. {g → 0.00444433}],
    {rbp, 0.0021, 0.004}, {L, 70, 150},
    ColorFunctionScaling → True,
    PlotLegends → Automatic,
    Contours → {Automatic, 71},
    ContourStyle → Thick,
    PerformanceGoal → "Quality",
    PlotRange → Full,
    PlotTheme → "Detailed",
    Frame → True, FrameStyle → Directive[12, Bold, Thick, Darker@Gray],
    FrameLabel → {"crossover rate", "mean tract length"},
    ImageSize → Automatic, PlotRangePadding → Automatic,
    Epilog → {
      Inset[Graphics[{Black, Text[Style["+", 14, Bold]]}],
        {0.002673853733400405, 113.24025414204982}]]],
  Graphics[
    {EdgeForm[{Thick, Dotted, Black}], FaceForm[], Rectangle[{0.0013, 185}, {0.0016, 227}]}]
]

```

