

```

SetDirectory[NotebookDirectory[]];

Study number of junctions supported by  $\geq K$  samples.

aggregatedJunctionCounts = Drop[Import["hg19.sample.stats.tsv", "TSV"], 1];

totalJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 2}]]];
annotatedJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 3}]]];
exonSkipJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 4}]]];
altStartEndJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 5}]]];
novelJunctions =
  exonSkips = Transpose[Transpose[aggregatedJunctionCounts][[{1, 6}]]];

Proportion of junctions in  $\geq 8k$  samples that are annotated

annotatedJunctions[[16 446 - 7999]]
{8000, 105 624}

annotatedJunctions[[16 446 - 7999]][[2]] / totalJunctions[[16 446 - 7999]][[2]] // N
0.997705

exonSkipAnnotatedJunctions = Transpose[{annotatedJunctions[[All, 1]],
  annotatedJunctions[[All, 2]] + exonSkipJunctions[[All, 2]]}];
someEvidenceJunctions = Transpose[{annotatedJunctions[[All, 1]],
  annotatedJunctions[[All, 2]] + exonSkipJunctions[[All, 2]] +
  altStartEndJunctions[[All, 2]]}];

mathematicaColors = ColorData[97, "ColorList"]
{█, █, █, █, █, █, █, █, █, █, █, █, █, █, █, █}

```

Introduce levels of evidence, and annotated with junction counts for ≥ 1000 samples

```

labelForm[x_, y___] := Text[Style[x, FontFamily → "Arial",
  FontSize → Scaled[.033], Bold, TextAlignment → Left], y]

idx = 16 446 - 999
15 447

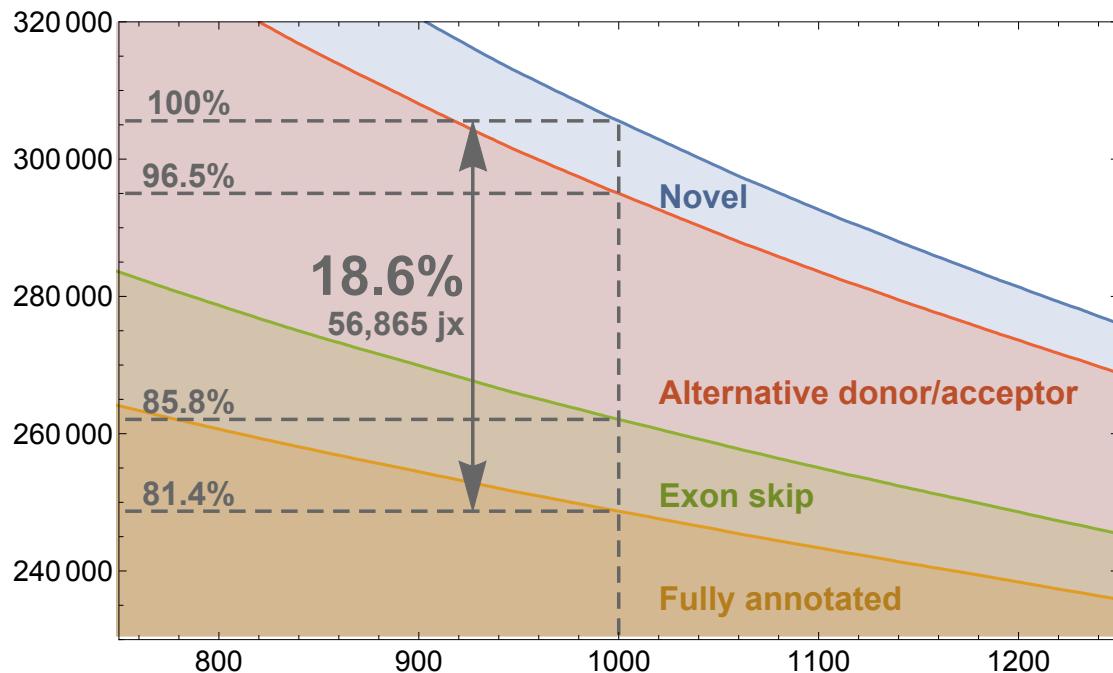
bigLabelForm[x_, y___] := Text[Style[x, FontFamily → "Arial",
  FontSize → Scaled[.055], Bold, TextAlignment → Right], y]

```

```

dashedColor = Darker[Gray, 0.2]; numberStartPos = 785; adjust = 2600;
insetAnnotationPlot = Show[ListPlot[{totalJunctions, annotatedJunctions,
    exonSkipAnnotatedJunctions, someEvidenceJunctions}, Joined -> True,
    PlotRange -> {{750, 1250}, {230 000, 320 000}}, Filling -> Axis, Frame -> True,
    ImageSize -> Large, BaseStyle -> {FontFamily -> "Arial", FontSize -> 15}],
Graphics[{Directive[Thickness[0.0035], Dashing[0.013], dashedColor],
    Line[{{1000, 0}, totalJunctions[[idx]]}],
    Line[{{0, totalJunctions[[idx]][[2]]}, totalJunctions[[idx]]}],
    Line[{{0, annotatedJunctions[[idx]][[2]]}, annotatedJunctions[[idx]]}],
    Line[{{0, exonSkipAnnotatedJunctions[[idx]][[2]]}, exonSkipAnnotatedJunctions[[idx]]}],
    Line[{{0, someEvidenceJunctions[[idx]][[2]]}, someEvidenceJunctions[[idx]]}],
    Directive[{Dashing[None], Arrowheads[{-0.05, 0.05}]}],
    Arrow[{{927, annotatedJunctions[[idx]][[2]]}, {927, totalJunctions[[idx]][[2]]}}],
    bigLabelForm[ToString[NumberForm[N[100 - annotatedJunctions[[idx, 2]]/totalJunctions[[idx, 2]]*100, 3],
        DigitBlock -> 3]] <> "%", {923, 283 000}, {1, 0}], labelForm[
    ToString[NumberForm[totalJunctions[[idx, 2]] - annotatedJunctions[[idx, 2]],
        DigitBlock -> 3]] <> " jx", {923, 276 000}, {1, 0}],
    labelForm["100%", {numberStartPos, totalJunctions[[idx]][[2]] + adjust}],
    labelForm[ToString[NumberForm[N[someEvidenceJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]]*100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, someEvidenceJunctions[[idx]][[2]] + adjust}],
    labelForm[ToString[NumberForm[N[annotatedJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]]*100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, annotatedJunctions[[idx]][[2]] + adjust}],
    labelForm[ToString[NumberForm[N[exonSkipAnnotatedJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]]*100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, exonSkipAnnotatedJunctions[[idx]][[2]] + adjust}],
    (*Directive[Thickness[0.0035], Arrowheads[.035], Dashing[None], dashedColor],
    Arrow[{{1000, totalJunctions[[idx]][[2]] + 41000},
        {1000, totalJunctions[[idx]][[2]] + 100}}], labelForm[ToString[NumberForm[N[100 - annotatedJunctions[[idx, 2]]/totalJunctions[[idx, 2]]*100, 3]] <>
        "% of jx unannotated, but\n" <> ToString[NumberForm[someEvidenceJunctions[[idx, 2]]/totalJunctions[[idx, 2]]*100/N, 3]] <>
        "% of jx have donor and/or\nacceptor site in annotation",
        {1005, 330000}, {-1, 0}], *)Darker[mathematicaColors[[1]], 0.2],
    labelForm["Novel", {1020, 294 500}, {-1, 0}],
    Darker[mathematicaColors[[4]], 0.2],
    labelForm["Alternative donor/acceptor", {1020, 266 000}, {-1, 0}],
    Darker[mathematicaColors[[3]], 0.2],
    labelForm["Exon skip", {1020, 251 000}, {-1, 0}],
    Darker[mathematicaColors[[2]], 0.2],
    labelForm["Fully annotated", {1020, 236 000}, {-1, 0}]})]

```



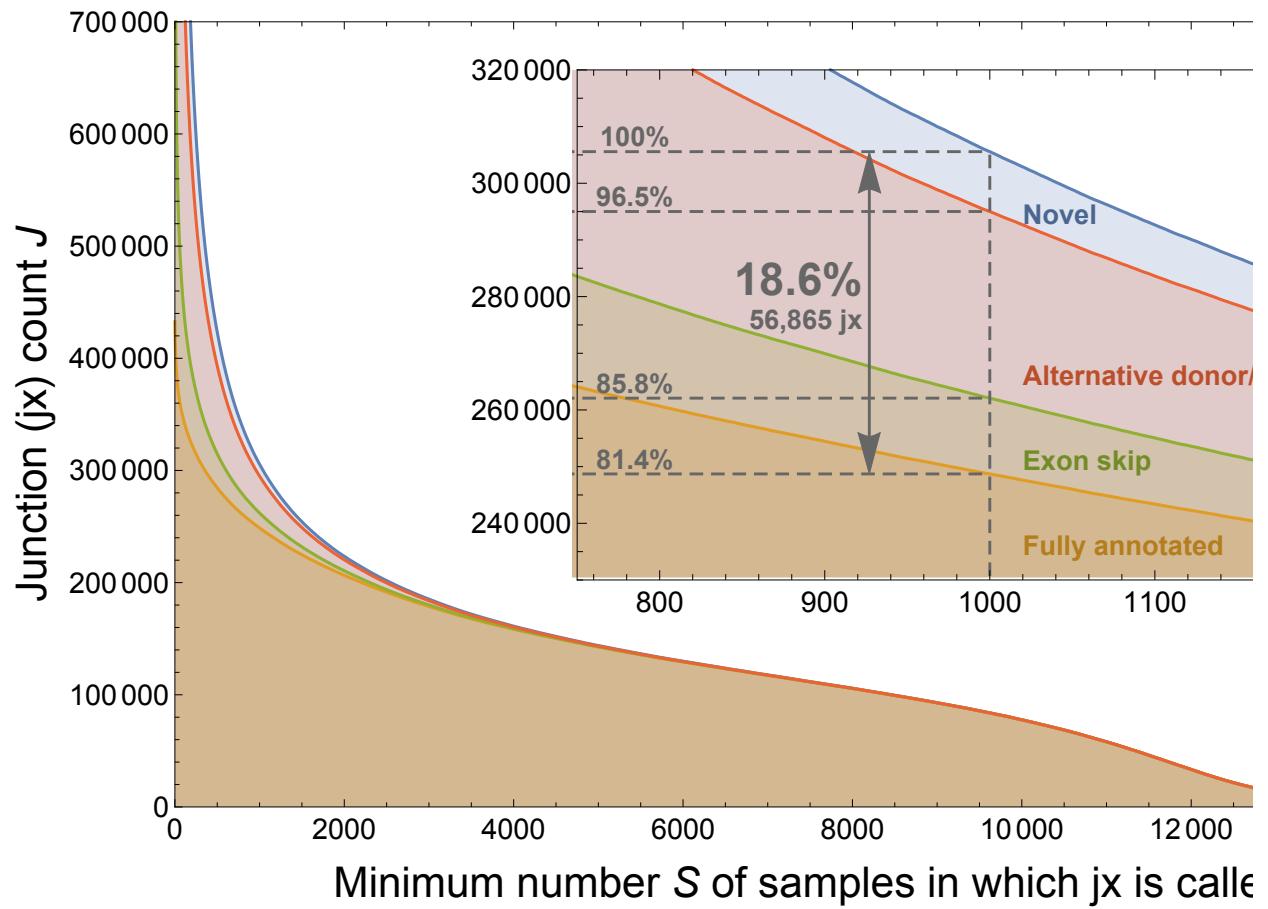
```
baseImageSize = {576, 408} * 1.3
```

```
{748.8, 530.4}
```

```

bigAnnotationPlot = ListPlot[{totalJunctions, annotatedJunctions,
  exonSkipAnnotatedJunctions, someEvidenceJunctions}, Joined → True,
  PlotRange → {{0, 15 000}, {0, 700 000}}, Filling → Axis, Frame → True,
  ImageSize → baseImageSize, BaseStyle → {FontFamily → "Arial", FontSize → 15},
  Epilog → Inset[insetAnnotationPlot, {9000, 420 000}, Automatic, 11000],
  FrameLabel → {Style["Minimum number  $S$  of samples in which jx is called", 22],
  Style["Junction (jx) count  $J$ ", 22]}]

```



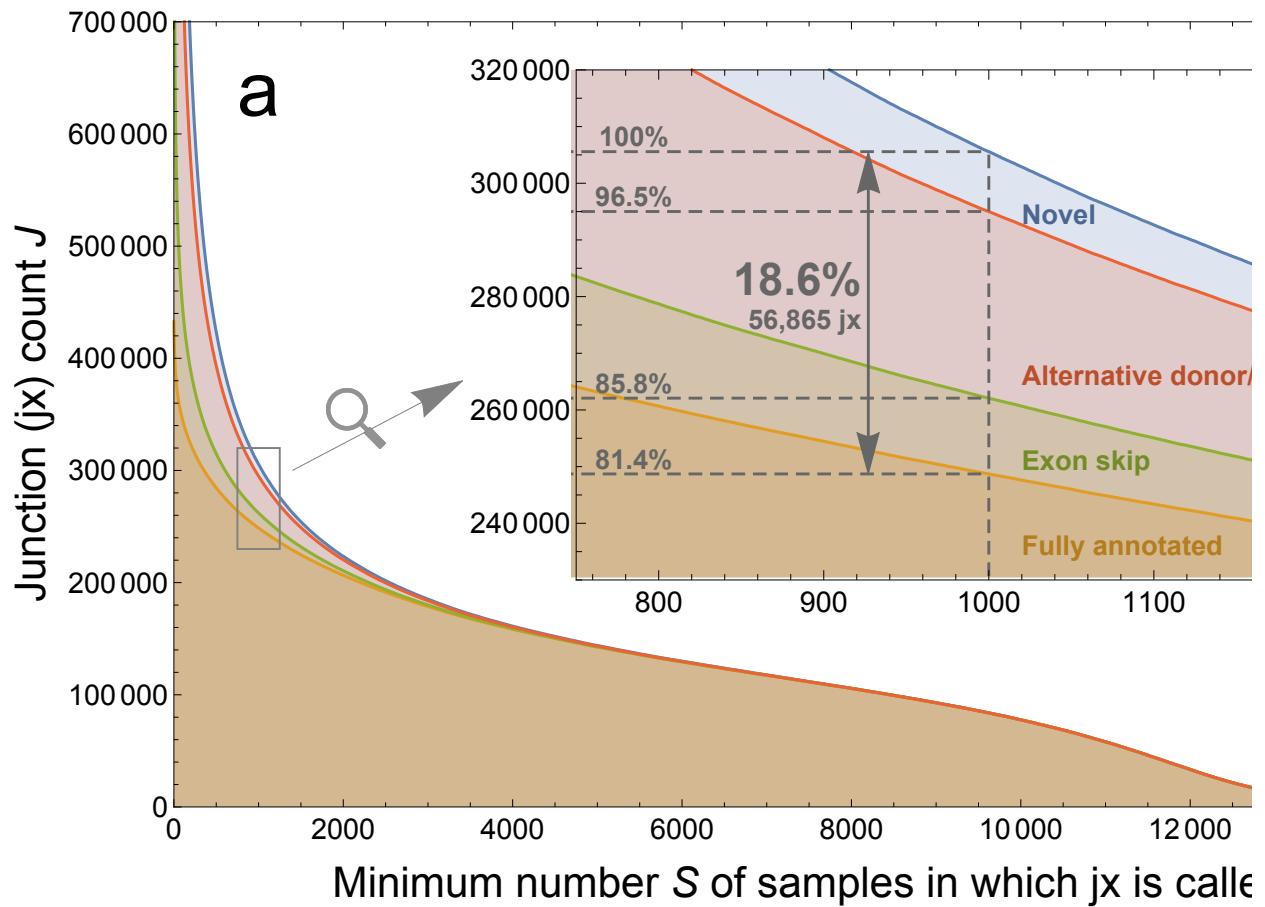
```
magnifyingGlass = Import["mag.png"]
```



```

fig1a = Show[bigAnnotationPlot,
  Graphics[{EdgeForm[Directive[Gray, Thickness[.0015]]],
    Transparent, Rectangle[{750, 230000}, {1250, 320000}]}],
  Graphics[{Gray, Arrow[{{1400, 300000}, {3410, 380000}}]}],
  Graphics[{Opacity[0.5], Inset[magnifyingGlass, {1800, 320000}, {0, 0}, 700]}],
  Graphics[{Black, Text[Style["a", FontFamily -> "Arial", FontSize -> 40],
    {1000, 640000}]}], ImageSize -> baseImageSize]

```



```

statsBySample = Drop[Import["!awk '$6>=100000' " <>
  NotebookDirectory[] <> "hg19.stats_by_sample.tsv", "TSV"], 1];

```

```
jxConsidered = Length[statsBySample]
```

```
10311
```

Overlaps by sample-->

```

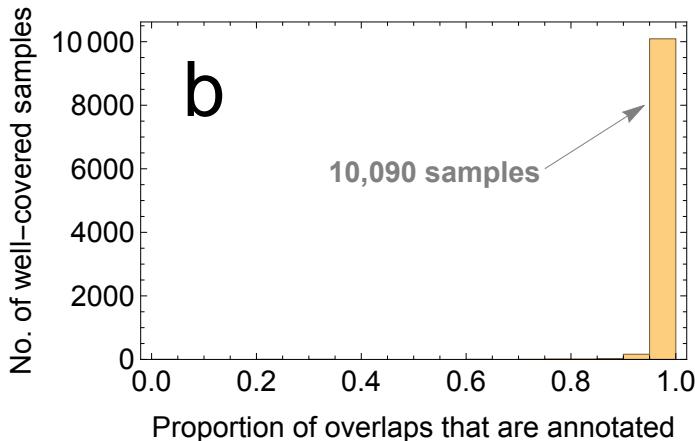
largerLabelForm[x_, y___] := Text[Style[x, FontFamily -> "Arial",
  FontSize -> Scaled[.053], Bold, TextAlignment -> Left], y]

```

```
jxHistList = HistogramList[statsBySample[[All, 11]] / statsBySample[[All, 10]], 15]
{{0, 1/20, 1/10, 3/20, 1/5, 1/4, 3/10, 7/20, 2/5, 9/20, 1/2, 11/20, 3/5, 13/20, 7/10, 3/4, 4/5, 17/20, 9/10, 19/20, 1}, {1, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2, 0, 2, 11, 10, 29, 163, 10090}}

lastBin = jxHistList[[2, 20]]
10090

padding = {{80, 10}, {60, 12}};
fig1b = Show[Histogram[statsBySample[[All, 11]] / statsBySample[[All, 10]] // N,
15, Frame → True, ImageSize → baseImageSize * .5,
BaseStyle → {FontFamily → "Arial", FontSize → 15}, ImagePadding → padding,
FrameLabel → {Style["Proportion of overlaps that are annotated", 15],
Style["No. of well-covered samples", 15]}],
Graphics[{Gray, Arrow[{{0.75, 6000}, {0.94, 8000}}}],
largerLabelForm[ToString[NumberForm[lastBin, DigitBlock -> 3]] <> " samples",
{0.54, 5900}]}, Graphics[
{Black, Text[Style["b", FontFamily → "Arial", FontSize → 40], {0.1, 8500}]}]]]
```

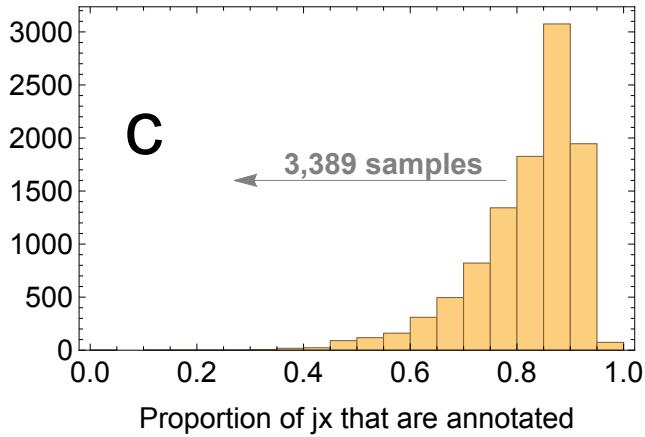


Junctions by sample-->

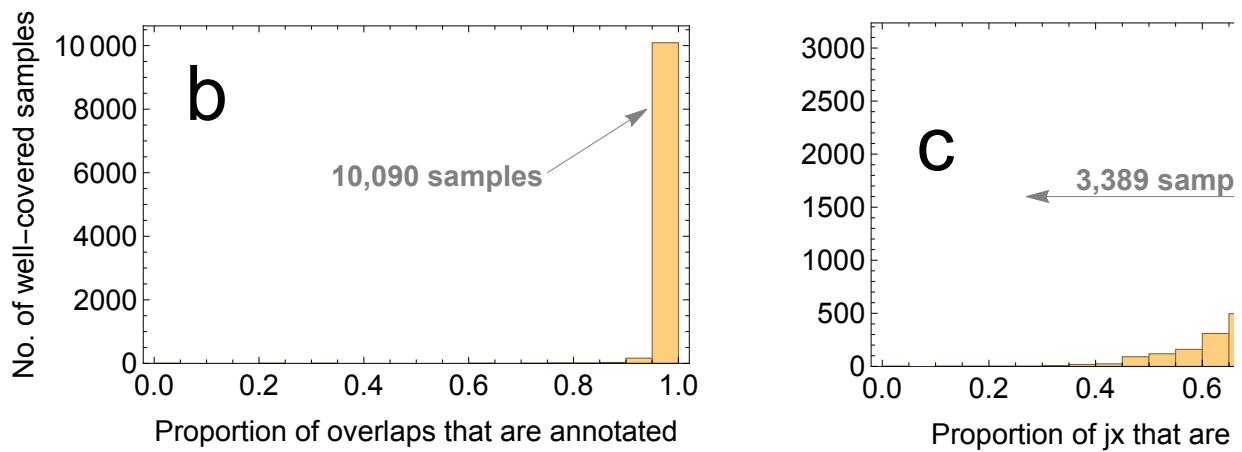
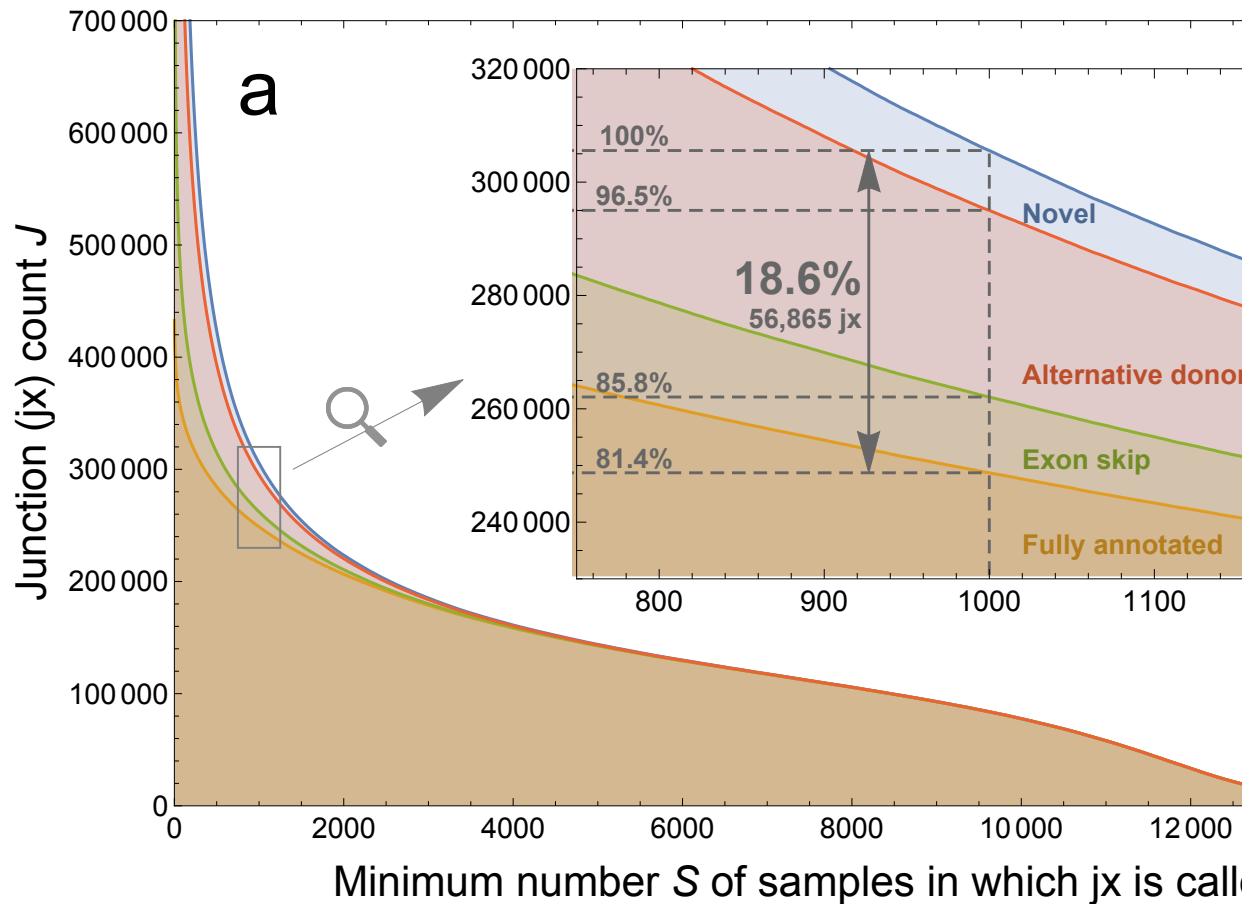
```
jxHistList2 = HistogramList[statsBySample[[All, 7]] / statsBySample[[All, 6]], 15]
{{0, 1/20, 1/10, 3/20, 1/5, 1/4, 3/10, 7/20, 2/5, 9/20, 1/2, 11/20, 3/5, 13/20, 7/10, 3/4, 4/5, 17/20, 9/10, 19/20, 1}, {1, 0, 1, 1, 1, 6, 18, 23, 90, 118, 160, 310, 496, 821, 1342, 1827, 3075, 1946, 74}]

lessThan80 = Total[jxHistList2[[2, Range[1, 16]]]]
3389
```

```
padding2 = {{75, 10}, {60, 12}};  
fig1c = Show[  
  Histogram[statsBySample[[All, 7]] / statsBySample[[All, 6]] // N,  
   15, Frame -> True, ImageSize -> baseImageSize *.5, ImagePadding -> padding2,  
   BaseStyle -> {FontFamily -> "Arial", FontSize -> 15},  
   FrameLabel -> {Style["Proportion of jx that are annotated", 15], None}],  
  Graphics[{Gray, Arrow[{{0.78, 1600}, {0.27, 1600}}]}],  
  largerLabelForm[ToString[NumberForm[lessThan80, DigitBlock -> 3]] <>  
   " samples", {0.55, 1760}]], Graphics[  
  {Black, Text[Style["c", FontFamily -> "Arial", FontSize -> 40], {0.1, 2100}]}]]
```



```
fig1 = Grid[{{fig1a}, {Grid[{{fig1b, fig1c}}]}}]
```



```
Export["jxannotation.pdf", fig1]
jxannotation.pdf

SEQC comparison: subset of 1720 samples out of the 21504 that were aligned by SEQC using magic,
rmake, subread. Shows that when a junction is in a lot of samples, it's found by other aligners.

aggregatedJunctionCounts = Drop[Import["hg19.seqc_sample.stats.tsv", "TSV"], 1];

totalJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 2}]]];
oneAlignerJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 6}]]];
twoAlignerJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 7}]]];
threeAlignerJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 8}]]];

totalJunctions[[1710 - 79]]
{80, 348531}

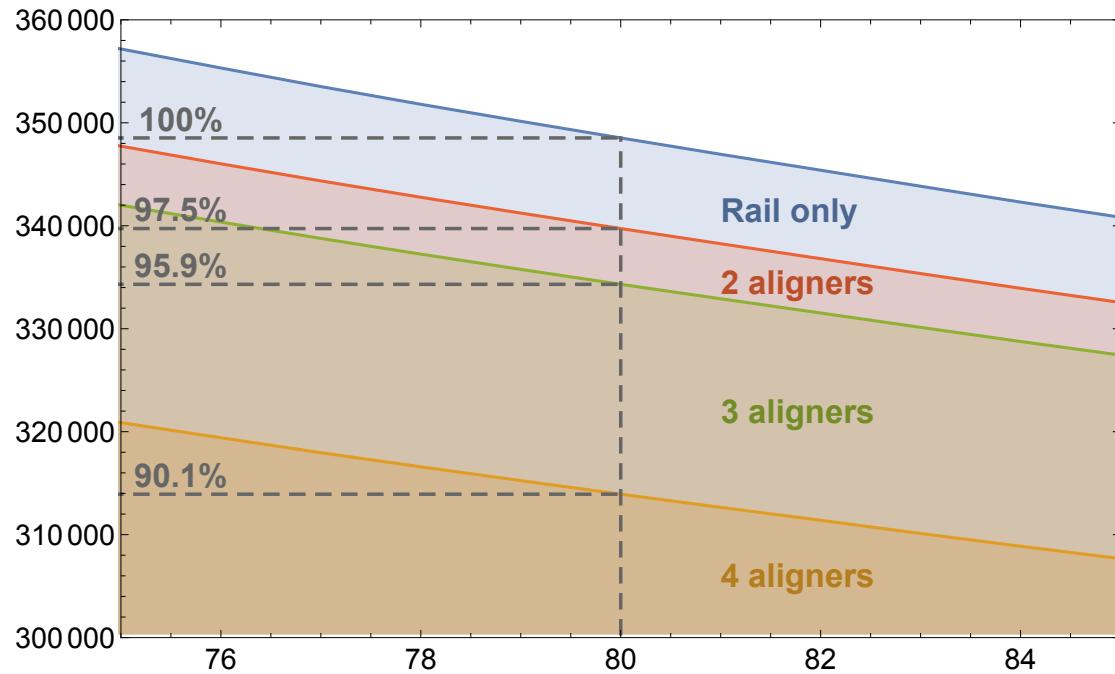
idx = 1710 - 79
1631

atLeastTwoAlignersJunctions = Transpose[{oneAlignerJunctions[[All, 1]],
  threeAlignerJunctions[[All, 2]] + twoAlignerJunctions[[All, 2]]}];
atLeastOneAlignerJunctions = Transpose[
 {oneAlignerJunctions[[All, 1]], oneAlignerJunctions[[All, 2]] +
  twoAlignerJunctions[[All, 2]] + threeAlignerJunctions[[All, 2]]}];
```

```

dashedColor = Darker[Gray, 0.2]; numberStartPos = 75.6;
insetAnnotationPlot = Show[ListPlot[{totalJunctions, threeAlignerJunctions,
    atLeastTwoAlignersJunctions, atLeastOneAlignerJunctions}, Joined -> True,
    PlotRange -> {{75, 85}, {300 000, 360 000}}, Filling -> Axis, Frame -> True,
    ImageSize -> Large, BaseStyle -> {FontFamily -> "Arial", FontSize -> 15}],
Graphics[{Directive[Thickness[0.0035], Dashing[0.013], dashedColor],
    Line[{{80, 0}, totalJunctions[[idx]]}],
    Line[{{0, totalJunctions[[idx]][[2]]}, totalJunctions[[idx]]}],
    Line[{{0, atLeastOneAlignerJunctions[[idx]][[2]]},
        atLeastOneAlignerJunctions[[idx]]}],
    Line[{{0, atLeastTwoAlignersJunctions[[idx]][[2]]},
        atLeastTwoAlignersJunctions[[idx]]}],
    Line[{{0, threeAlignerJunctions[[idx]][[2]]}, threeAlignerJunctions[[idx]]}],
    Directive[{Dashing[None], Arrowheads[{-0.03, 0.03}]}],
    (*Arrow[{{79.63, atLeastOneAlignerJunctions[[idx]][[2]]},
        {79.63, totalJunctions[[idx]][[2]]}}], bigLabelForm[ToString[NumberForm[
        N[100-atLeastOneAlignerJunctions[[idx,2]]/totalJunctions[[idx,2]]*100,3],
        DigitBlock->3]]<>"%",{79.5, 345500}, {1,0}], labelForm[
        ToString[NumberForm[totalJunctions[[idx,2]]-atLeastOneAlignerJunctions[[idx,2]],DigitBlock->3]]<>" jx", {79.5, 341500}, {1,0}],*),
    labelForm["100%", {numberStartPos, totalJunctions[[idx]][[2]] + 1800}],
    labelForm[ToString[NumberForm[N[atLeastOneAlignerJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]] * 100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, atLeastOneAlignerJunctions[[idx]][[2]] + 1800}],
    labelForm[ToString[NumberForm[N[threeAlignerJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]] * 100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, threeAlignerJunctions[[idx]][[2]] + 1800}],
    labelForm[ToString[NumberForm[N[atLeastTwoAlignersJunctions[[idx, 2]]/
        totalJunctions[[idx, 2]] * 100, 3], DigitBlock -> 3]] <> "%",
        {numberStartPos, atLeastTwoAlignersJunctions[[idx]][[2]] + 1800}],
    (*Directive[Thickness[0.0035], Arrowheads[.035], Dashing[None], dashedColor],
    Arrow[{{1000, totalJunctions[[idx]][[2]]+41000},
        {1000, totalJunctions[[idx]][[2]]+100}}], labelForm[ToString[NumberForm[
        N[100-annotatedJunctions[[idx,2]]/totalJunctions[[idx,2]]*100],3]]<>
        "% of jx unannotated, but\n">>ToString[NumberForm[
        someEvidenceJunctions[[idx,2]]/totalJunctions[[idx,2]]*100//N,3]]]<>
        "% of jx have donor and/or\\nacceptor site in annotation",
        {1005, 330000}, {-1,0}],*)Darker[mathematicaColors[[1]], 0.2],
    labelForm["Rail only", {81, 341500}, {-1, 0}],
    Darker[mathematicaColors[[4]], 0.2],
    labelForm["2 aligners", {81, 334500}, {-1, 0}],
    Darker[mathematicaColors[[3]], 0.2],
    labelForm["3 aligners", {81, 322000}, {-1, 0}],
    Darker[mathematicaColors[[2]], 0.2],
    labelForm["4 aligners", {81, 306000}, {-1, 0}]})]

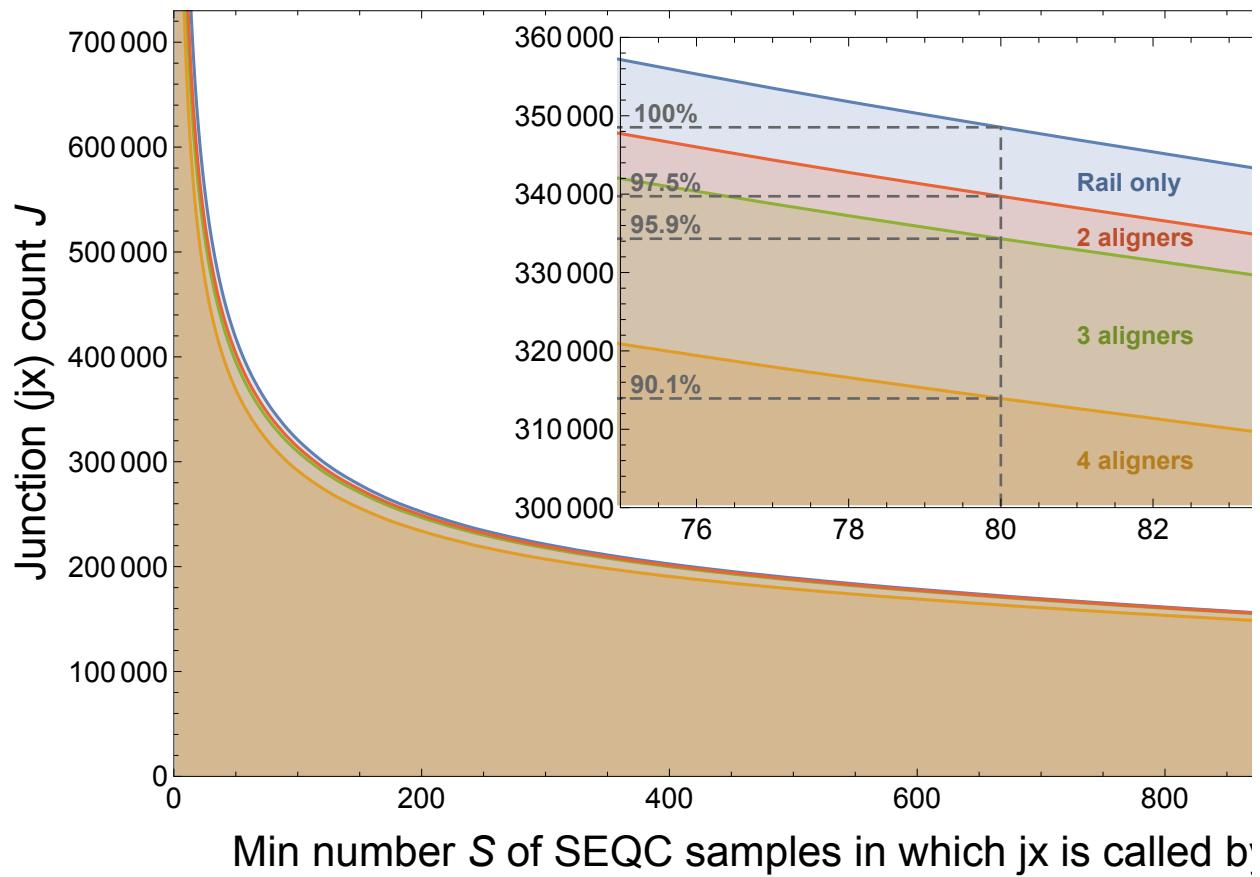
```



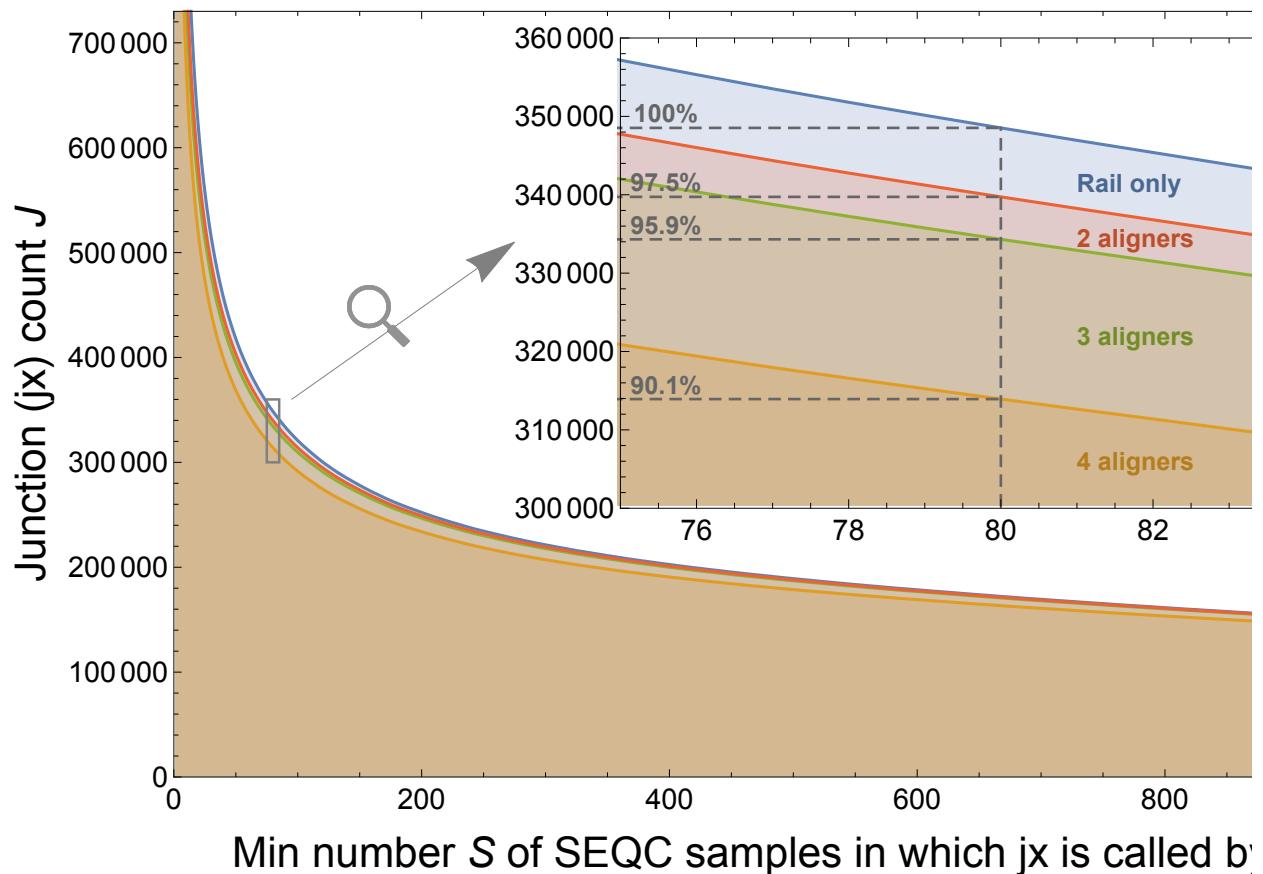
```

seqcPlot = ListPlot[{totalJunctions, threeAlignerJunctions,
  atLeastTwoAlignersJunctions, atLeastOneAlignerJunctions}, Joined -> True,
  PlotRange -> {{0, 1000}, {0, 730 000}}, Filling -> Axis, Frame -> True,
  ImageSize -> baseImageSize, BaseStyle -> {FontFamily -> "Arial", FontSize -> 15},
  Epilog -> Inset[insetAnnotationPlot, {625, 470 000}, Automatic, 700], FrameLabel ->
  {Style["Min number  $S$  of SEQC samples in which  $j_x$  is called by Rail", 22],
  Style["Junction ( $j_x$ ) count  $J$ ", 22}]}

```



```
suppfigseqc = Show[seqcPlot, Graphics[{EdgeForm[Directive[Gray, Thickness[.002]]],  
Transparent, Rectangle[{75, 300000}, {85, 360000}]}],  
Graphics[{Gray, Arrow[{{95, 360000}, {275, 510000}}]}],  
Graphics[{Opacity[0.5], Inset[magnifyingGlass, {140, 410000}, {0, 0}, 50]}],  
ImageSize → baseImageSize]
```



```
Export["seqc.pdf", suppfigseqc]
```

```
seqc.pdf
```

Repeat Figure 1a, except at project level for supplement.

```
aggregatedJunctionCounts = Drop[Import["hg19.project.stats.tsv", "TSV"], 1];  
  
totalJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 2}]]];  
annotatedJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 3}]]];  
exonSkipJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 4}]]];  
altStartEndJunctions = Transpose[Transpose[aggregatedJunctionCounts][[{1, 5}]]];  
novelJunctions =  
exonSkips = Transpose[Transpose[aggregatedJunctionCounts][[{1, 6}]]];
```

```

exonSkipAnnotatedJunctions = Transpose[{annotatedJunctions[[All, 1]],
    annotatedJunctions[[All, 2]] + exonSkipJunctions[[All, 2]]}];
someEvidenceJunctions = Transpose[{annotatedJunctions[[All, 1]],
    annotatedJunctions[[All, 2]] + exonSkipJunctions[[All, 2]] +
    altStartEndJunctions[[All, 2]]}];

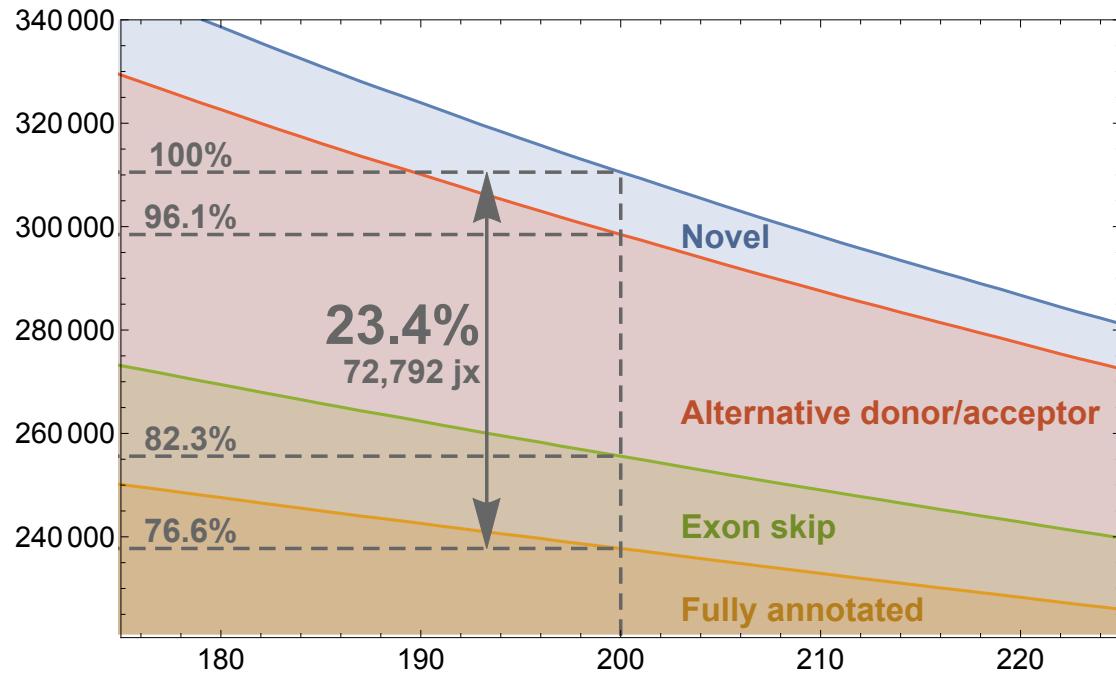
Introduce levels of evidence, and annotated with junction counts for ≥ 30 samples.

totalJunctions[[784 - 199]]
{200, 310 536}

idx = 784 - 199
585

dashedColor = Darker[Gray, 0.2]; numberStartPos = 178.5; adjust = 3400;
insetAnnotationPlot = Show[ListPlot[{totalJunctions, annotatedJunctions,
    exonSkipAnnotatedJunctions, someEvidenceJunctions}, Joined → True,
    PlotRange → {{175, 225}, {220 500, 340 000}}, Filling → Axis, Frame → True,
    ImageSize → Large, BaseStyle → {FontFamily → "Arial", FontSize → 15}],
    Graphics[{Directive[Thickness[0.0035], Dashing[0.013], dashedColor],
        Line[{{200, 0}, totalJunctions[[idx]]}],
        Line[{{0, totalJunctions[[idx]][[2]]}, totalJunctions[[idx]]}],
        Line[{{0, annotatedJunctions[[idx]][[2]]}, annotatedJunctions[[idx]]}],
        Line[{{0, exonSkipAnnotatedJunctions[[idx]][[2]]},
            exonSkipAnnotatedJunctions[[idx]]}],
        Line[{{0, someEvidenceJunctions[[idx]][[2]]}, someEvidenceJunctions[[idx]]}],
        Directive[{Dashing[None], Arrowheads[{-0.05, .05}]}],
        Arrow[{{193.3, annotatedJunctions[[idx]][[2]]},
            {193.3, totalJunctions[[idx]][[2]]}}], bigLabelForm[ToString[NumberForm[
                N[100 - annotatedJunctions[[idx, 2]] / totalJunctions[[idx, 2]] * 100, 3],
                DigitBlock → 3]] <> "%", {193, 281 000}, {1, 0}], labelForm[
                ToString[NumberForm[totalJunctions[[idx, 2]] - annotatedJunctions[[idx, 2]],
                    DigitBlock → 3]] <> " jx", {193, 272 000}, {1, 0}],
        labelForm["100%", {numberStartPos, totalJunctions[[idx]][[2]] + adjust}],
        labelForm[ToString[NumberForm[N[someEvidenceJunctions[[idx, 2]]] /
            totalJunctions[[idx, 2]] * 100, 3], DigitBlock → 3]] <> "%",
        {numberStartPos, someEvidenceJunctions[[idx]][[2]] + adjust}],
        labelForm[ToString[NumberForm[N[annotatedJunctions[[idx, 2]]] /
            totalJunctions[[idx, 2]] * 100, 3], DigitBlock → 3]] <> "%",
        {numberStartPos, annotatedJunctions[[idx]][[2]] + adjust}],
        labelForm[ToString[NumberForm[N[exonSkipAnnotatedJunctions[[idx, 2]]] /
            totalJunctions[[idx, 2]] * 100, 3], DigitBlock → 3]] <> "%",
        {numberStartPos, exonSkipAnnotatedJunctions[[idx]][[2]] + adjust}],
        Darker[mathematicaColors[[1]], 0.2],
        labelForm["Novel", {203, 298 000}, {-1, 0}],
        Darker[mathematicaColors[[4]], 0.2],
        labelForm["Alternative donor/acceptor", {203, 264 000}, {-1, 0}],
        Darker[mathematicaColors[[3]], 0.2],
        labelForm["Exon skip", {203, 242 000}, {-1, 0}],
        Darker[mathematicaColors[[2]], 0.2],
        labelForm["Fully annotated", {203, 226 000}, {-1, 0}]}]
    }]
]

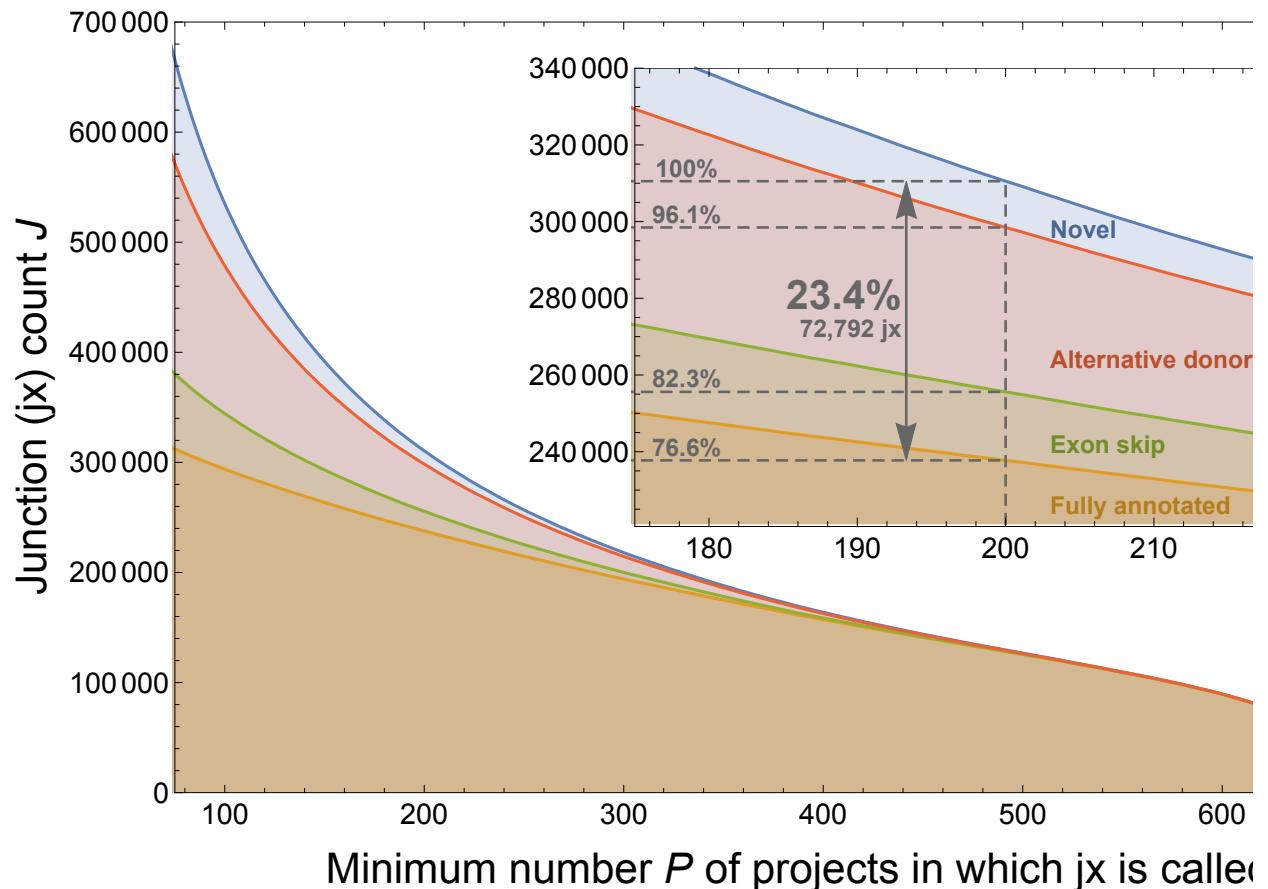
```



```

bigAnnotationPlot = ListPlot[{totalJunctions,
  annotatedJunctions, Transpose[{Transpose[annotatedJunctions][[1]],
    Transpose[annotatedJunctions][[2]] + Transpose[exonSkipJunctions][[2]]}],
  Transpose[{Transpose[annotatedJunctions][[1]],
    Transpose[annotatedJunctions][[2]] + Transpose[exonSkipJunctions][[2]] +
    Transpose[altStartEndJunctions][[2]]}],
  Joined -> True, PlotRange -> {{75, 700}, {0, 700 000}}, Filling -> Axis,
  Frame -> True, ImageSize -> baseImageSize,
  BaseStyle -> {FontFamily -> "Arial", FontSize -> 15},
  Epilog -> Inset[insetAnnotationPlot, {465, 440 000}, Automatic, 425],
  FrameLabel -> {Style["Minimum number  $P$  of projects in which jx is called", 22],
  Style["Junction (jx) count  $J$ ", 22]}]

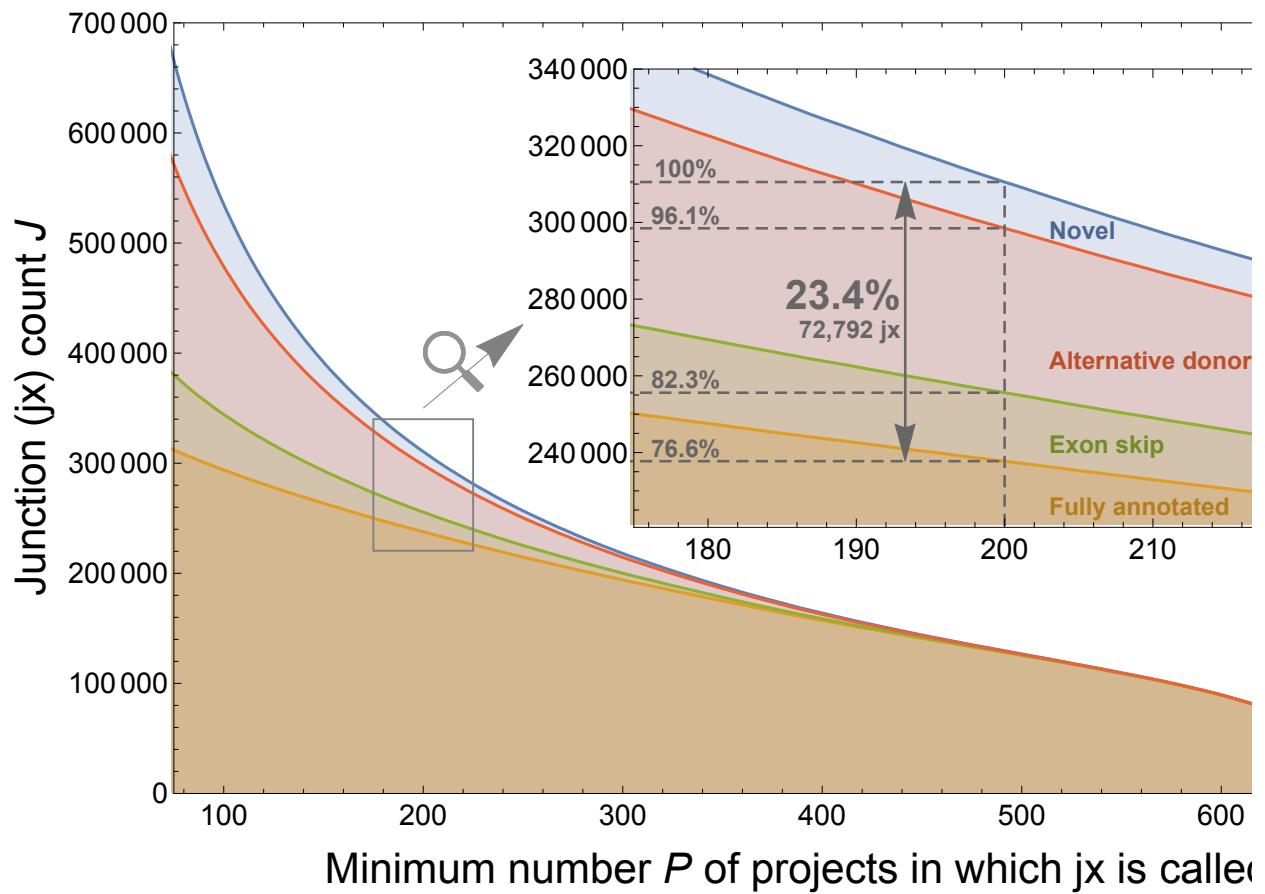
```



```

suppfigproj =
  Show[bigAnnotationPlot, Graphics[{EdgeForm[Directive[Gray, Thickness[.0015]]],
  Transparent, Rectangle[{175, 220500}, {225, 340000}]}], 
  Graphics[{Gray, Arrow[{{200, 350000}, {250, 426000}}]}], 
  Graphics[{Opacity[0.5], Inset[magnifyingGlass, {200, 365000}, {0, 0}, 30]}], 
  ImageSize -> baseImageSize]

```



```
Export["projlevel.pdf", suppfigproj]
```

```
projlevel.pdf
```

For how many runs are we missing Biosample submission dates? We ran the command
`cat index_to_SRA_accession.tsv | grep -vwFf <(cat biosample_tags.tsv | cut-f10 | tail-n+2)`
`>missing_biosample_dates.tsv`

in the sra/hg19 directory of the repo nellore/runs to obtain that dates were missing for only
 $77/21504=0.3\%$ of runs. Our analysis is reasonably complete if we ignore them.

```

junctionsEvidenceVsDatesGeq20 = Drop[
  Import["!gzip -cd hg19.sample_count_submission_date_overlap_geq_20.tsv.gz",
  "TSV"], 1];

```

```

junctionsEvidenceVsDatesGeq[x_, y_: junctionsEvidenceVsDatesGeq20] :=
  Select[y, #[[1]] ≥ x &];
talliedJunctionsGeq[x_, y_: junctionsEvidenceVsDatesGeq20] :=
  SortBy[Tally[junctionsEvidenceVsDatesGeq[x, y][[All, 4]]], First];
accumulatedJunctionsGeq[x_, y_: junctionsEvidenceVsDatesGeq20] :=
  (talliedJunctions = talliedJunctionsGeq[x, y];
   Transpose[{talliedJunctions[[All, 1]], Accumulate[talliedJunctions[[All, 2]]]}])

```

Convert from days after 2/27/2009 to dates.

```

Clear[daysToDate]

daysToDate[x_] := DatePlus[DateObject[{2009, 02, 27}], x]

```

When were junctions supported by reads in $\geq 20, 40, 80, 160$ reads across samples found?

```

twentyThirteen = 1404; daysToDate[twentyThirteen]

```

 Tue 1 Jan 2013

```

dateFormat = {"Month", "/", "Day", "/", "YearShort"};

```

Design ticks to intersect 1/1/2013.

```

dateTicks = ({#,DateString[daysToDate[#], dateFormat]} & /@ Range[0, 2070, 234])
{{0, 02/27/09}, {234, 10/19/09}, {468, 06/10/10}, {702, 01/30/11}, {936, 09/21/11},
 {1170, 05/12/12}, {1404, 01/01/13}, {1638, 08/23/13}, {1872, 04/14/14}}

```

```

lastDay = Max[junctionsEvidenceVsDatesGeq20[[All, 4]]]

```

2070

```

dateTicks =
 Append[dateTicks, {lastDay, DateString[daysToDate[lastDay], dateFormat]}]
{{0, 02/27/09}, {234, 10/19/09}, {468, 06/10/10},
 {702, 01/30/11}, {936, 09/21/11}, {1170, 05/12/12}, {1404, 01/01/13},
 {1638, 08/23/13}, {1872, 04/14/14}, {2070, 10/29/14}}

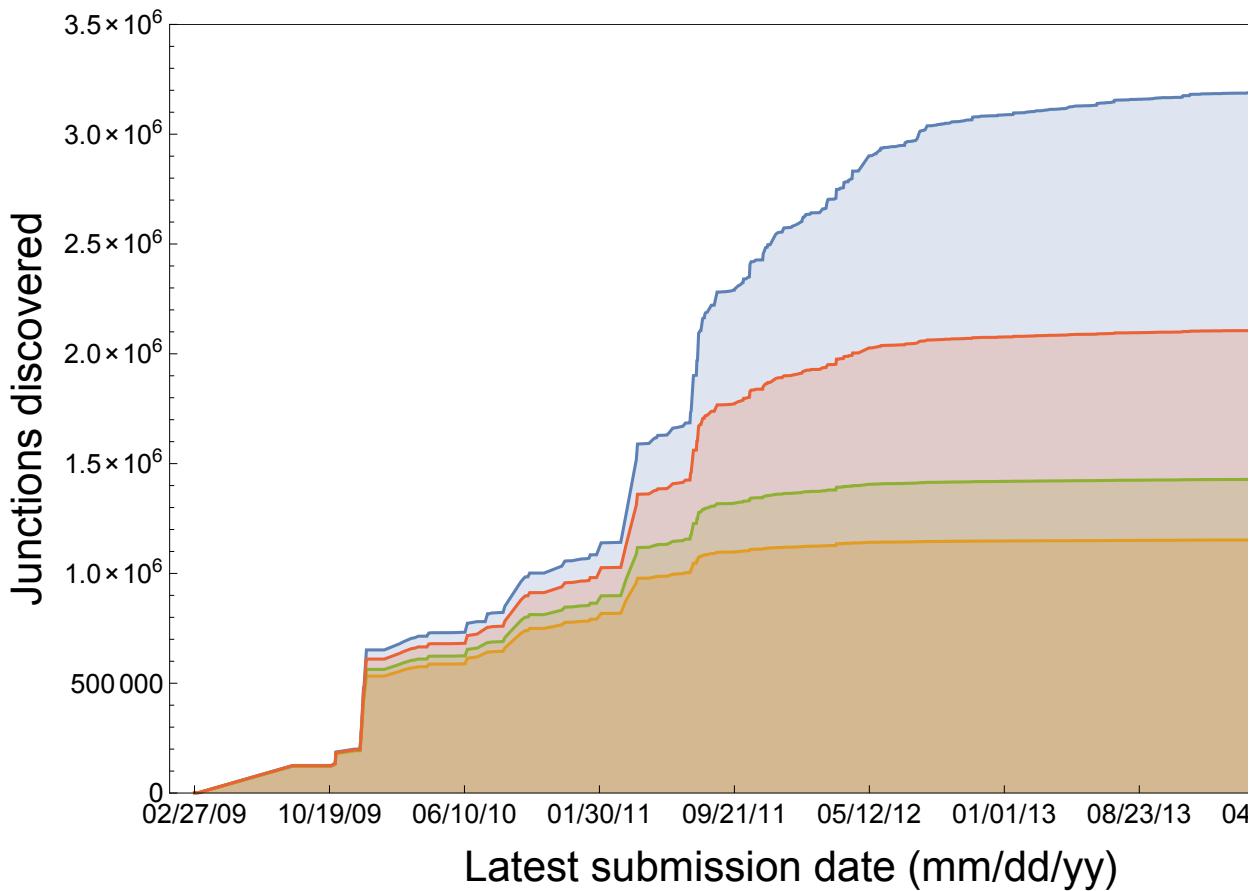
```

```

baseJunctionsPlotData = accumulatedJunctionsGeq /@ {20, 120, 80, 40};

```

```
baseJunctionsPlot = ListPlot[baseJunctionsPlotData, Joined → True, Filling → Axis,
  Frame → True, FrameTicks → {{Automatic, None}, {dateTicks, None}},
  BaseStyle → {FontFamily → "Arial", FontSize → 14},
  FrameLabel → {Style["Latest submission date (mm/dd/yy)", 22],
  Style["Junctions discovered", 22]},
  ImageSize → baseImageSize, PlotRange → {All, {0, 3.5 * 10^6}}]
```



```
sortedDays = Sort[junctionsEvidenceVsDatesGeg20[[All, 4]]];
junctionsInCommons = Count[sortedDays, #] & /@ Commonest[sortedDays, 7]
{123 759, 124 121, 155 069, 163 007, 124 664, 252 628, 162 196}
```

These correspond to, respectively....

```
daysToDate /@ Commonest[sortedDays, 7]
```

```
{Sun 16 Aug 2009, Mon 14 Dec 2009, Thu 17 Dec 2009,
Tue 22 Dec 2009, Thu 17 Mar 2011, Mon 4 Apr 2011, Tue 12 Jul 2011}
```

Some of these dates correspond to jumps in the plot above. Grepping for the submission dates in biosample_tags.tsv in sra/hg19 gives samples in the following projects:

1. who cares
2. who cares
3. Study of 69 LCLs (2)(Understanding mechanisms underlying human gene expression variation with RNA sequencing, by Pickrell et al.) (SRP001540) 17 Dec 2009
4. Study of 41 Coriell cell lines (SRP001563)(Polymorphic cis-and trans-regulation of human gene expression, by Cheung et al.) 22 Dec 2009
5. Illumina bodyMap2 (ERP000546) 17-Mar-2011
6. University of Washington Human Reference Epigenome Mapping Project (SRP001371)(total RNA, fetal tissues, contributed most junctions on a single day (4 April 2011)); note also that on this day, there are two more projects: SRP005309, a microRNA study with negligible # junctions, and SRP005846, for which grepping hg19.stats_by_sample.tsv gives ~95 annotated jx, < 50k each of 4 samples. So overwhelmingly dominant contribution on 4 April 2011 is UW.
7. ENCODE long RNA-seq from CSHL (SRP007461) 12-Jul-2011

Annotate plot with the top 5 projects (3,4,5,6, and 7 above); the 1st, 2nd, and 5th are about the same size but have many fewer junctions than the top 4 contributors. However, bodyMap 2 is interesting in that GENCODE incorporated it in its annotation.

Find GEUVADIS. Grepping biosample_tags.tsv gives that the GEUVADIS submission date was 2012-11-07. This was

```
daysToDate[1349]
```

 Wed 7 Nov 2012

A glance at the tallies below shows that just 11294 novel junctions were contributed on the day GEUVADIS samples were submitted to Biosample!

```
tallied = Tally[sortedDays]
```

```
{ {0, 1110}, {7, 624}, {170, 123 759}, {213, 70}, {237, 55}, {244, 10 389}, {245, 50 729}, {278, 13 334}, {287, 9}, {290, 124 121}, {293, 155 069}, {294, 9318}, {298, 163 007}, {329, 50}, {356, 28 624}, {362, 8285}, {376, 15 841}, {382, 2878}, {388, 6900}, {398, 10}, {403, 72}, {406, 14 113}, {411, 1632}, {418, 42}, {448, 214}, {454, 582}, {469, 1429}, {473, 40 933}, {474, 15}, {490, 7270}, {504, 1}, {508, 36 019}, {511, 5}, {515, 3186}, {525, 1521}, {535, 1267}, {538, 27 738}, {566, 112 496}, {573, 21 129}, {578, 695}, {581, 17 055}, {606, 49}, {637, 31 778}, {642, 22 678}, {648, 1386}, {654, 35}, {671, 8394}, {677, 434}, {679, 790}, {684, 2408}, {686, 15 188}, {697, 26}, {705, 54 568}, {718, 692}, {739, 1470}, {748, 124 664}, {766, 252 628}, {768, 70 376}, {788, 2783}, {791, 8257}, {795, 10 214}, {798, 6030}, {801, 1370}, {803, 10 521}, {819, 1655}, {824, 14 227}, {829, 17 430}, {837, 3048}, {847, 5902}, {851, 14 186}, {859, 904}, {860, 47 163}, {861, 5641}, {865, 162 196}, {870, 1600}, {871, 67 018}, {872, 1659}, {874, 123 474}, {877, 11 239}, {878, 268}, {881, 54 855}, {882, 2501}, {884, 819}, {885, 18 340}, {886, 4513}, {889, 2528}, {896, 31 640}, {901, 548}, {906, 59 520}, {910, 86}, {921, 1682}, {930, 3243}, {935, 3954}, {936, 1}, {937, 5905}, {943, 14 686}, {945, 1035}, {950, 11 778}, {951, 337}, {952, 16 763}, {956, 566}, {959, 5763}, {962, 3082}, {963, 57 275}, {965, 11 806}, {969, 594}, {970, 943}, {971, 1358}, {972, 3666}, {974, 1507}, {985, 462}, {986, 27 558}, {990, 28 199}, {993, 1969}, {994, 11 304}, {998, 70}, {1007, 44 673}, {1008, 3680}, {1011, 6130}, {1014, 2180}, {1015, 86}, {1018, 657}, {1019, 701}, {1021, 15 130}, {1022, 3414}, {1033, 1845}, {1035, 1188}, {1036, 3752}, {1043, 7398}, {1047, 5465}, {1049, 3837}, {1053, 5246}, {1054, 12 145}, {1055, 6846}, {1056, 3}, {1057, 1475}, {1059, 8659}, {1060, 26}, {1061, 3282}, {1067, 946}, }
```

```

{1069, 6340}, {1070, 4}, {1081, 519}, {1083, 1}, {1084, 686}, {1089, 16982},
{1092, 1}, {1095, 2970}, {1096, 19603}, {1097, 9063}, {1098, 10301}, {1099, 1719},
{1102, 4}, {1104, 396}, {1106, 193}, {1109, 4}, {1112, 3385}, {1113, 40568},
{1116, 455}, {1118, 942}, {1119, 3927}, {1120, 869}, {1124, 596}, {1125, 1235},
{1126, 23732}, {1129, 3015}, {1132, 100}, {1133, 69}, {1134, 7020}, {1140, 7364},
{1141, 34226}, {1146, 92}, {1151, 259}, {1166, 57205}, {1169, 11585}, {1174, 1557},
{1175, 663}, {1180, 7802}, {1181, 420}, {1182, 4}, {1183, 4788}, {1188, 10953},
{1189, 204}, {1190, 9213}, {1193, 123}, {1194, 34}, {1195, 2713}, {1197, 200},
{1200, 206}, {1202, 146}, {1207, 2681}, {1208, 398}, {1209, 338}, {1214, 19},
{1215, 42}, {1222, 4291}, {1224, 528}, {1225, 378}, {1228, 1}, {1230, 243},
{1231, 65}, {1232, 11969}, {1236, 5507}, {1238, 24}, {1239, 500}, {1242, 210},
{1243, 860}, {1244, 19}, {1245, 893}, {1250, 2128}, {1253, 8601}, {1257, 30849},
{1258, 3488}, {1263, 3301}, {1264, 515}, {1265, 211}, {1267, 3047}, {1270, 16862},
{1277, 651}, {1279, 20}, {1281, 1433}, {1285, 1131}, {1286, 5}, {1287, 552},
{1288, 2187}, {1289, 97}, {1290, 219}, {1291, 205}, {1292, 111}, {1293, 5},
{1294, 72}, {1295, 2060}, {1296, 29}, {1299, 40}, {1300, 818}, {1301, 1947},
{1302, 45}, {1305, 61}, {1307, 196}, {1308, 326}, {1309, 255}, {1313, 5697},
{1315, 109}, {1316, 383}, {1321, 316}, {1326, 864}, {1327, 252}, {1330, 1712},
{1333, 659}, {1334, 1000}, {1335, 189}, {1337, 1451}, {1339, 1613}, {1340, 22},
{1344, 37}, {1348, 961}, {1349, 11294}, {1350, 142}, {1351, 1460}, {1352, 33},
{1357, 494}, {1361, 301}, {1362, 1604}, {1364, 1204}, {1365, 132}, {1369, 1},
{1370, 29}, {1371, 445}, {1372, 115}, {1376, 229}, {1379, 455}, {1386, 156},
{1387, 212}, {1388, 59}, {1389, 70}, {1392, 68}, {1393, 2766}, {1398, 188},
{1399, 2}, {1400, 293}, {1405, 676}, {1406, 13}, {1407, 463}, {1410, 330},
{1411, 87}, {1417, 70}, {1418, 936}, {1419, 19}, {1420, 6790}, {1421, 2}, {1425, 39},
{1431, 625}, {1433, 28}, {1439, 400}, {1443, 3072}, {1446, 25}, {1447, 692},
{1452, 1233}, {1453, 42}, {1456, 344}, {1459, 2966}, {1461, 116}, {1463, 16},
{1469, 626}, {1475, 2617}, {1476, 277}, {1477, 40}, {1480, 842}, {1482, 1675},
{1484, 543}, {1485, 7}, {1487, 160}, {1488, 177}, {1492, 2}, {1495, 329},
{1496, 783}, {1497, 27}, {1498, 56}, {1501, 29}, {1502, 1028}, {1503, 632},
{1508, 174}, {1510, 136}, {1512, 1857}, {1516, 5111}, {1518, 954}, {1522, 911},
{1523, 405}, {1524, 633}, {1526, 1298}, {1529, 743}, {1531, 23}, {1533, 56},
{1537, 1}, {1539, 493}, {1540, 116}, {1543, 191}, {1545, 127}, {1546, 15},
{1551, 175}, {1552, 427}, {1553, 12}, {1556, 241}, {1557, 413}, {1558, 246},
{1559, 62}, {1560, 191}, {1561, 559}, {1564, 68}, {1565, 7893}, {1567, 304},
{1568, 87}, {1571, 1651}, {1574, 26}, {1575, 127}, {1578, 63}, {1579, 1071},
{1580, 53}, {1581, 78}, {1582, 589}, {1585, 27}, {1586, 16}, {1587, 499}, {1588, 11},
{1589, 1}, {1592, 2609}, {1594, 37}, {1595, 7294}, {1596, 299}, {1597, 263},
{1599, 364}, {1600, 87}, {1603, 12}, {1606, 11}, {1607, 289}, {1610, 25}, {1612, 6},
{1613, 199}, {1614, 194}, {1615, 47}, {1617, 79}, {1621, 854}, {1622, 174},
{1623, 1005}, {1627, 21}, {1628, 1}, {1629, 116}, {1633, 2}, {1634, 119},
{1636, 590}, {1642, 42}, {1643, 279}, {1644, 1}, {1645, 366}, {1646, 24}, {1648, 7},
{1649, 105}, {1650, 282}, {1651, 20}, {1655, 30}, {1656, 218}, {1657, 920},
{1660, 59}, {1663, 899}, {1665, 1533}, {1670, 911}, {1671, 16}, {1672, 4},
{1673, 731}, {1676, 39}, {1678, 864}, {1693, 2}, {1694, 270}, {1697, 578},
{1699, 6}, {1701, 367}, {1704, 7}, {1705, 8}, {1707, 15}, {1708, 31}, {1711, 1},
{1712, 1981}, {1713, 125}, {1714, 5338}, {1715, 190}, {1718, 54}, {1719, 1},
{1721, 54}, {1722, 41}, {1725, 74}, {1726, 5625}, {1728, 282}, {1729, 202},
{1730, 4}, {1733, 34}, {1740, 3}, {1741, 9}, {1743, 35}, {1744, 1355}, {1747, 360},
{1748, 227}, {1750, 21}, {1753, 10}, {1755, 82}, {1756, 52}, {1757, 96}, {1760, 6},
{1761, 71}, {1763, 166}, {1767, 471}, {1776, 133}, {1777, 1}, {1778, 18}, {1781, 92},
{1782, 811}, {1783, 36}, {1784, 106}, {1787, 5}, {1790, 3}, {1791, 11}, {1792, 311},
{1795, 19}, {1796, 329}, {1797, 12}, {1799, 14}, {1805, 37}, {1806, 172},
{1810, 14}, {1811, 40}, {1813, 4}, {1817, 1}, {1818, 8}, {1819, 5}, {1820, 14},
{1822, 4}, {1823, 42}, {1824, 14}, {1825, 1383}, {1830, 35}, {1831, 9}, {1837, 129},

```

```
{1838, 59}, {1840, 1037}, {1841, 79}, {1842, 2}, {1844, 8}, {1846, 126}, {1847, 44}, {1849, 835}, {1850, 3305}, {1851, 2376}, {1852, 9}, {1853, 242}, {1854, 20}, {1855, 6}, {1859, 96}, {1860, 336}, {1861, 13}, {1862, 10}, {1865, 60}, {1866, 43}, {1867, 13}, {1868, 44}, {1869, 1}, {1872, 23}, {1873, 19}, {1874, 15}, {1875, 14}, {1880, 5724}, {1881, 26}, {1884, 9}, {1886, 1}, {1889, 7}, {1890, 1}, {1894, 23}, {1895, 42}, {1896, 94}, {1897, 7}, {1900, 3}, {1901, 3}, {1902, 7}, {1904, 302}, {1908, 4}, {1909, 495}, {1910, 183}, {1915, 176}, {1918, 2}, {1921, 707}, {1928, 2}, {1929, 6}, {1930, 266}, {1931, 13}, {1933, 92}, {1937, 9}, {1938, 7}, {1949, 5}, {1950, 3}, {1958, 1}, {1960, 3}, {1966, 62}, {1967, 1}, {1970, 36}, {1974, 3}, {1975, 5140}, {1977, 1}, {1993, 6}, {2010, 1}, {2014, 1}, {2029, 9}, {2070, 358}}
```

Find GEUV day's rank:

```
Reverse[SortBy[tallied, Last]]
```

```
{766, 252 628}, {298, 163 007}, {865, 162 196}, {293, 155 069}, {748, 124 664}, {290, 124 121}, {170, 123 759}, {874, 123 474}, {566, 112 496}, {768, 70 376}, {871, 67 018}, {906, 59 520}, {963, 57 275}, {1166, 57 205}, {881, 54 855}, {705, 54 568}, {245, 50 729}, {860, 47 163}, {1007, 44 673}, {473, 40 933}, {1113, 40 568}, {508, 36 019}, {1141, 34 226}, {637, 31 778}, {896, 31 640}, {1257, 30 849}, {356, 28 624}, {990, 28 199}, {538, 27 738}, {986, 27 558}, {1126, 23 732}, {642, 22 678}, {573, 21 129}, {1096, 19 603}, {885, 18 340}, {829, 17 430}, {581, 17 055}, {1089, 16 982}, {1270, 16 862}, {952, 16 763}, {376, 15 841}, {686, 15 188}, {1021, 15 130}, {943, 14 686}, {824, 14 227}, {851, 14 186}, {406, 14 113}, {278, 13 334}, {1054, 12 145}, {1232, 11 969}, {965, 11 806}, {950, 11 778}, {1169, 11 585}, {994, 11 304}, {1349, 11 294}, {877, 11 239}, {1188, 10 953}, {803, 10 521}, {244, 10 389}, {1098, 10 301}, {795, 10 214}, {294, 9318}, {1190, 9213}, {1097, 9063}, {1059, 8659}, {1253, 8601}, {671, 8394}, {362, 8285}, {791, 8257}, {1565, 7893}, {1180, 7802}, {1043, 7398}, {1140, 7364}, {1595, 7294}, {490, 7270}, {1134, 7020}, {388, 6900}, {1055, 6846}, {1420, 6790}, {1069, 6340}, {1011, 6130}, {798, 6030}, {937, 5905}, {847, 5902}, {959, 5763}, {1880, 5724}, {1313, 5697}, {861, 5641}, {1726, 5625}, {1236, 5507}, {1047, 5465}, {1714, 5338}, {1053, 5246}, {1975, 5140}, {1516, 5111}, {1183, 4788}, {886, 4513}, {1222, 4291}, {935, 3954}, {1119, 3927}, {1049, 3837}, {1036, 3752}, {1008, 3680}, {972, 3666}, {1258, 3488}, {1022, 3414}, {1112, 3385}, {1850, 3305}, {1263, 3301}, {1061, 3282}, {930, 3243}, {515, 3186}, {962, 3082}, {1443, 3072}, {837, 3048}, {1267, 3047}, {1129, 3015}, {1095, 2970}, {1459, 2966}, {382, 2878}, {788, 2783}, {1393, 2766}, {1195, 2713}, {1207, 2681}, {1475, 2617}, {1592, 2609}, {889, 2528}, {882, 2501}, {684, 2408}, {1851, 2376}, {1288, 2187}, {1014, 2180}, {1250, 2128}, {1295, 2060}, {1712, 1981}, {993, 1969}, {1301, 1947}, {1512, 1857}, {1033, 1845}, {1099, 1719}, {1330, 1712}, {921, 1682}, {1482, 1675}, {872, 1659}, {819, 1655}, {1571, 1651}, {411, 1632}, {1339, 1613}, {1362, 1604}, {870, 1600}, {1174, 1557}, {1665, 1533}, {525, 1521}, {974, 1507}, {1057, 1475}, {739, 1470}, {1351, 1460}, {1337, 1451}, {1281, 1433}, {469, 1429}, {648, 1386}, {1825, 1383}, {801, 1370}, {971, 1358}, {1744, 1355}, {1526, 1298}, {535, 1267}, {1125, 1235}, {1452, 1233}, {1364, 1204}, {1035, 1188}, {1285, 1131}, {0, 1110}, {1579, 1071}, {1840, 1037}, {945, 1035}, {1502, 1028}, {1623, 1005}, {1334, 1000}, {1348, 961}, {1518, 954}, {1067, 946}, {970, 943}, {1118, 942}, {1418, 936}, {1657, 920}, {1670, 911}, {1522, 911}, {859, 904}, {1663, 899}, {1245, 893}, {1120, 869}, {1678, 864}, {1326, 864}, {1243, 860}, {1621, 854}, {1480, 842}, {1849, 835}, {884, 819}, {1300, 818}, {1782, 811}, {679, 790}, {1496, 783}, {1529, 743}, {1673, 731}, {1921, 707}, {1019, 701}, {578, 695}, {1447, 692}, {718, 692}, {1084, 686}, {1405, 676}, {1175, 663}, {1333, 659}, {1018, 657}, {1277, 651}, {1524, 633}, {1503, 632}, {1469, 626}, {1431, 625}, {7, 624}, {1124, 596}, {969, 594}, {1636, 590}, {1582, 589}, {454, 582}, {1697, 578}, {956, 566}, {1561, 559}, {1287, 552}, {901, 548}, {1484, 543}, {1224, 528}, {1081, 519}, {1264, 515}, {1239, 500}, {1587, 499}, {1909, 495},
```

```

{1357, 494}, {1539, 493}, {1767, 471}, {1407, 463}, {985, 462}, {1379, 455},
{1116, 455}, {1371, 445}, {677, 434}, {1552, 427}, {1181, 420}, {1557, 413},
{1523, 405}, {1439, 400}, {1208, 398}, {1104, 396}, {1316, 383}, {1225, 378},
{1701, 367}, {1645, 366}, {1599, 364}, {1747, 360}, {2070, 358}, {1456, 344},
{1209, 338}, {951, 337}, {1860, 336}, {1410, 330}, {1796, 329}, {1495, 329},
{1308, 326}, {1321, 316}, {1792, 311}, {1567, 304}, {1904, 302}, {1361, 301},
{1596, 299}, {1400, 293}, {1607, 289}, {1728, 282}, {1650, 282}, {1643, 279},
{1476, 277}, {1694, 270}, {878, 268}, {1930, 266}, {1597, 263}, {1151, 259},
{1309, 255}, {1327, 252}, {1558, 246}, {1230, 243}, {1853, 242}, {1556, 241},
{1376, 229}, {1748, 227}, {1290, 219}, {1656, 218}, {448, 214}, {1387, 212},
{1265, 211}, {1242, 210}, {1200, 206}, {1291, 205}, {1189, 204}, {1729, 202},
{1197, 200}, {1613, 199}, {1307, 196}, {1614, 194}, {1106, 193}, {1560, 191},
{1543, 191}, {1715, 190}, {1335, 189}, {1398, 188}, {1910, 183}, {1488, 177},
{1915, 176}, {1551, 175}, {1622, 174}, {1508, 174}, {1806, 172}, {1763, 166},
{1487, 160}, {1386, 156}, {1202, 146}, {1350, 142}, {1510, 136}, {1776, 133},
{1365, 132}, {1837, 129}, {1575, 127}, {1545, 127}, {1846, 126}, {1713, 125},
{1193, 123}, {1634, 119}, {1629, 116}, {1540, 116}, {1461, 116}, {1372, 115},
{1292, 111}, {1315, 109}, {1784, 106}, {1649, 105}, {1132, 100}, {1289, 97},
{1859, 96}, {1757, 96}, {1896, 94}, {1933, 92}, {1781, 92}, {1146, 92}, {1600, 87},
{1568, 87}, {1411, 87}, {1015, 86}, {910, 86}, {1755, 82}, {1841, 79}, {1617, 79},
{1581, 78}, {1725, 74}, {1294, 72}, {403, 72}, {1761, 71}, {1417, 70}, {1389, 70},
{998, 70}, {213, 70}, {1133, 69}, {1564, 68}, {1392, 68}, {1231, 65}, {1578, 63},
{1966, 62}, {1559, 62}, {1305, 61}, {1865, 60}, {1838, 59}, {1660, 59}, {1388, 59},
{1533, 56}, {1498, 56}, {237, 55}, {1721, 54}, {1718, 54}, {1580, 53}, {1756, 52},
{329, 50}, {606, 49}, {1615, 47}, {1302, 45}, {1868, 44}, {1847, 44}, {1866, 43},
{1895, 42}, {1823, 42}, {1642, 42}, {1453, 42}, {1215, 42}, {418, 42}, {1722, 41},
{1811, 40}, {1477, 40}, {1299, 40}, {1676, 39}, {1425, 39}, {1805, 37}, {1594, 37},
{1344, 37}, {1970, 36}, {1783, 36}, {1830, 35}, {1743, 35}, {654, 35}, {1733, 34},
{1194, 34}, {1352, 33}, {1708, 31}, {1655, 30}, {1501, 29}, {1370, 29}, {1296, 29},
{1433, 28}, {1585, 27}, {1497, 27}, {1881, 26}, {1574, 26}, {1060, 26}, {697, 26},
{1610, 25}, {1446, 25}, {1646, 24}, {1238, 24}, {1894, 23}, {1872, 23}, {1531, 23},
{1340, 22}, {1750, 21}, {1627, 21}, {1854, 20}, {1651, 20}, {1279, 20}, {1873, 19},
{1795, 19}, {1419, 19}, {1244, 19}, {1214, 19}, {1778, 18}, {1671, 16}, {1586, 16},
{1463, 16}, {1874, 15}, {1707, 15}, {1546, 15}, {474, 15}, {1875, 14}, {1824, 14},
{1820, 14}, {1810, 14}, {1799, 14}, {1931, 13}, {1867, 13}, {1861, 13}, {1406, 13},
{1797, 12}, {1603, 12}, {1553, 12}, {1791, 11}, {1606, 11}, {1588, 11}, {1862, 10},
{1753, 10}, {398, 10}, {2029, 9}, {1937, 9}, {1884, 9}, {1852, 9}, {1831, 9},
{1741, 9}, {287, 9}, {1844, 8}, {1818, 8}, {1705, 8}, {1938, 7}, {1902, 7},
{1897, 7}, {1889, 7}, {1704, 7}, {1648, 7}, {1485, 7}, {1993, 6}, {1929, 6},
{1855, 6}, {1760, 6}, {1699, 6}, {1612, 6}, {1949, 5}, {1819, 5}, {1787, 5},
{1293, 5}, {1286, 5}, {511, 5}, {1908, 4}, {1822, 4}, {1813, 4}, {1730, 4},
{1672, 4}, {1182, 4}, {1109, 4}, {1102, 4}, {1070, 4}, {1974, 3}, {1960, 3},
{1950, 3}, {1901, 3}, {1900, 3}, {1790, 3}, {1740, 3}, {1056, 3}, {1928, 2},
{1918, 2}, {1842, 2}, {1693, 2}, {1633, 2}, {1492, 2}, {1421, 2}, {1399, 2},
{2014, 1}, {2010, 1}, {1977, 1}, {1967, 1}, {1958, 1}, {1890, 1}, {1886, 1},
{1869, 1}, {1817, 1}, {1777, 1}, {1719, 1}, {1711, 1}, {1644, 1}, {1628, 1},
{1589, 1}, {1537, 1}, {1369, 1}, {1228, 1}, {1092, 1}, {1083, 1}, {936, 1}, {504, 1}}

```

Position[Reverse[SortBy[tallied, Last]], {1349, 11294}]

{55}

GEUV is at 55!

```

geuvDate = 1349
1349

arrowLabelForm [x_, y___] := Text[Style[x, FontFamily -> "Arial",
  FontSize -> Scaled[.03], Bold, TextAlignment -> Left], y]

smallerLabelForm [x_, y___] := Text[Style[x, FontFamily -> "Arial",
  FontSize -> Scaled[.02], Bold, TextAlignment -> Left], y]

biggerLabelForm [x_, y___] := Text[Style[x, FontFamily -> "Arial",
  FontSize -> Scaled[.04], Bold, TextAlignment -> Left], y]

altLabelForm [x_, y___] := Text[Style[x, FontFamily -> "Arial",
  FontSize -> Scaled[.045], Bold, TextAlignment -> Left], y]

accJunc = accumulatedJunctionsGeq[20];

maxAtTwentyThirteen = Select[accJunc, #[[1]] <= twentyThirteen &][[-1]][[2]]
3 087 471

maxAtEnd = accJunc[[-1]][[2]]
3 211 228

```

How many junctions covered by ≥ 20 reads are there? Should agree with maxAtEnd.

```

Length[junctionsEvidenceVsDatesGeq20]
3 211 228

```

From the command line and in the runs/sra directory, run

```

join -2 3 <(cut -f10,11 hg19/biosample_tags.tsv | tail -n +2 | cut -dT' -f1 | sort-k1,1) <(sort -k3,3 intropo-
lis.idmap.v1.hg19.tsv) | awk '$2 < "2013-01-01"' | wc -l

```

to get that 7426 samples are before 2013, and change the $<$ to a \geq in the awk command to get that 14801 samples are ≥ 2013 . The 77 missing samples don't have Biosample submission dates, and they're ignored.

```

before2013 = 7426; after2013 = 14 001;

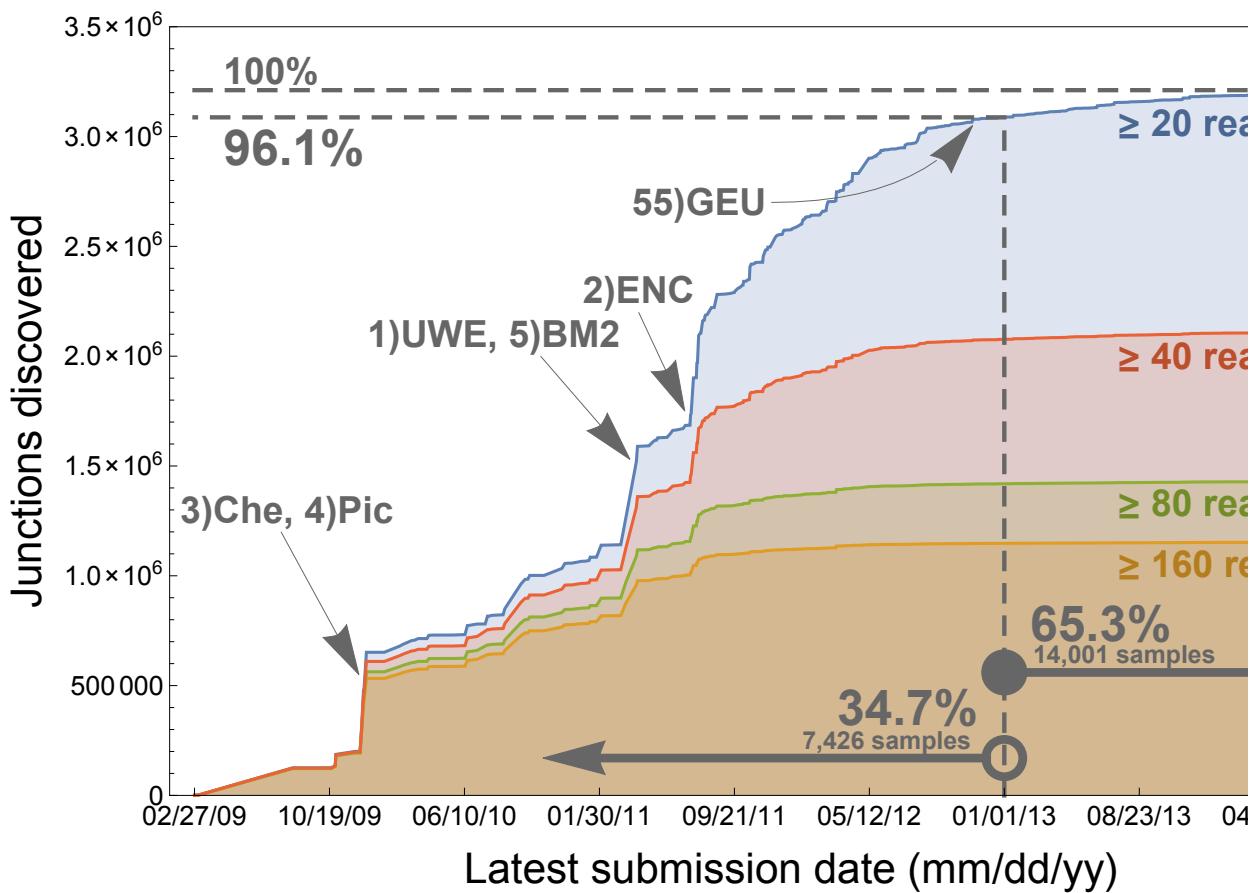
before2013 / (before2013 + after2013) // N
0.346572

```

```

labelColor = Darker[Gray, 0.2]; leftPos = 1600;
botPos = 110000; fig2 = Show[baseJunctionsPlot,
  Graphics[{labelColor, Arrow[{{150, 1.2 * 10^6}, {285, 550000}}], arrowLabelForm[
    "3)Che, 4)Pic", {160, 1.3 * 10^6}], Arrow[{{600, 2 * 10^6}, {755, 1.53 * 10^6}}],
  arrowLabelForm["1)UWE, 5)BM2", {600, 2.1 * 10^6}],
  Arrow[{{770, 2.2 * 10^6}, {850, 1.75 * 10^6}}],
  Arrow[BezierCurve[{{1000, 2.7 * 10^6}, {1249, 2.7 * 10^6}, {1349, 3.05 * 10^6}}]],
  arrowLabelForm["2)ENC", {770, 2.3 * 10^6}], Directive[Thickness[0.0035],
  Dashing[0.013], labelColor], arrowLabelForm["55)GEU", {875, 2.7 * 10^6}],
  Directive[Thickness[0.0035], Dashing[0.013], labelColor],
  Line[{{twentyThirteen, 0}, {twentyThirteen, maxAtTwentyThirteen}}],
  Line[{{twentyThirteen, maxAtTwentyThirteen}, {0, maxAtTwentyThirteen}}],
  Line[{{lastDay, maxAtEnd}, {lastDay, 0}}],
  Line[{{0, maxAtEnd}, {lastDay, maxAtEnd}}],
  arrowLabelForm["100%", {50, 3.3 * 10^6}, {-1, 0}], biggerLabelForm[ToString[
    NumberForm[N[maxAtTwentyThirteen / maxAtEnd * 100, 3], DigitBlock -> 3]] <> "%",
  {50, 2.95 * 10^6}, {-1, 0}], Darker[mathematicaColors[[2]], 0.2],
  arrowLabelForm["≥ 160 reads", {leftPos, 1.055 * 10^6}, {-1, 0}],
  Darker[mathematicaColors[[3]], 0.2],
  arrowLabelForm["≥ 80 reads", {leftPos, 1.33 * 10^6}, {-1, 0}],
  Darker[mathematicaColors[[4]], 0.2], arrowLabelForm["≥ 40 reads",
  {leftPos, 2 * 10^6}, {-1, 0}], Darker[mathematicaColors[[1]], 0.2],
  arrowLabelForm["≥ 20 reads", {leftPos, 3.06 * 10^6}, {-1, 0}],
  Dashing[None], Thickness[.007], Darker[Gray, .2], Arrowheads[{{0, .05}}],
  smallerLabelForm[ToString[NumberForm[after2013, DigitBlock -> 3]] <> " samples",
  {twentyThirteen + 50, botPos + 530000}, {-1, 0}],
  biggerLabelForm[ToString[NumberForm[
    N[after2013 / (after2013 + before2013) * 100, 3], DigitBlock -> 3]] <> "%",
  {twentyThirteen + 45, botPos + 670000}, {-1, 0}],
  smallerLabelForm[ToString[NumberForm[before2013, DigitBlock -> 3]] <>
  " samples", {twentyThirteen - 350, 249000}, {-1, 0}],
  biggerLabelForm[ToString[NumberForm[N[before2013 / (after2013 + before2013) *
  100, 3], DigitBlock -> 3]] <> "%", {twentyThirteen - 290, 400000},
  {-1, 0}], Arrow[{{twentyThirteen + 18, botPos + 450000},
  {twentyThirteen + 610, botPos + 450000}}],
  Disk[{twentyThirteen, botPos + 450000}, {40, 105000}],
  Arrow[{{twentyThirteen - 23, 170000}, {twentyThirteen - 800, 170000}}],
  Circle[{twentyThirteen, 170000}, {32, 85000}]]]

```



```
Export["dateplot.pdf", fig2]
```

```
dateplot.pdf
```

Format of next list is {GENCODE index, date}.

```
earliestGencodes =
  {#[[2]], #[[1, 1, 1]]} & /@ Select[{Position[#[[Range[5, 22]]], 1], #[[4]]} & /@
    junctionsEvidenceVsDatesGeq20, Length[#[[1]]] > 0 &];
```

Freeze dates taken from <http://www.gencodegenes.org/releases/>.

```
daysAfterDate [y_] := DateDifference[DateObject[{2009, 2, 27}], y]
```

```
gencodeFreezeDates =
  {DateObject[{2009, 7}], DateObject[{2009, 7}], DateObject[{2010, 1}],
   DateObject[{2010, 4}], DateObject[{2010, 11}], DateObject[{2010, 12}],
   DateObject[{2011, 3}], DateObject[{2011, 5}], DateObject[{2011, 7}],
   DateObject[{2011, 10}], DateObject[{2011, 12}], DateObject[{2012, 3}],
   DateObject[{2012, 6}], DateObject[{2012, 8}], DateObject[{2012, 11}],
   DateObject[{2013, 2}], DateObject[{2013, 4}], DateObject[{2013, 7}]};
```

```

gencodeAppearDates =
  {DateObject[{2009, 9}], DateObject[{2010, 3}], DateObject[{2010, 5}],
   DateObject[{2010, 11}], DateObject[{2011, 2}], DateObject[{2011, 4}],
   DateObject[{2011, 6}], DateObject[{2011, 9}], DateObject[{2011, 12}],
   DateObject[{2012, 2}], DateObject[{2012, 5}], DateObject[{2012, 7}],
   DateObject[{2012, 10}], DateObject[{2013, 1}], DateObject[{2013, 4}],
   DateObject[{2013, 6}], DateObject[{2013, 9}], DateObject[{2013, 12}]};

gencodeFreezeDays = QuantityMagnitude /@ daysAfterDate /@ gencodeFreezeDates
{124, 124, 308, 398, 612, 642, 732, 793, 854,
 946, 1007, 1098, 1190, 1251, 1343, 1435, 1494, 1585}

gencodeAppearDays = QuantityMagnitude /@ daysAfterDate /@ gencodeAppearDates
{186, 367, 428, 612, 704, 763, 824, 916, 1007,
 1069, 1159, 1220, 1312, 1404, 1494, 1555, 1647, 1738}

appearDateFormat = {"Month", "/", "YearShort"};

gencodeAppearDateTicks =
  {#,DateString[daysToDate[#], appearDateFormat]} & /@ gencodeAppearDays
{{186, 09/09}, {367, 03/10}, {428, 05/10}, {612, 11/10},
 {704, 02/11}, {763, 04/11}, {824, 06/11}, {916, 09/11}, {1007, 12/11},
 {1069, 02/12}, {1159, 05/12}, {1220, 07/12}, {1312, 10/12},
 {1404, 01/13}, {1494, 04/13}, {1555, 06/13}, {1647, 09/13}, {1738, 12/13}};

discoveryDaysToGencodeDays =
  {#[[1]], gencodeAppearDays[[#[[2]]]]} & /@ earliestGencodes;

toAcc = SortBy[Tally[#[[1]] & /@ earliestGencodes], First];
accumulatedAnnotated = Transpose[{toAcc[[All, 1]], Accumulate[toAcc[[All, 2]]]}];

baseImageSize
{748.8, 530.4}

toBoxWhisker = Table[#[[1]] & /@ Select[discoveryDaysToGencodeDays, #[[2]] == i &],
  {i, gencodeAppearDays}];

toBar = Length /@ toBoxWhisker
{251810, 1145, 10593, 5478, 7188, 9319, 3455,
 3119, 2579, 4879, 2088, 3196, 4599, 3265, 553, 515, 542, 667}

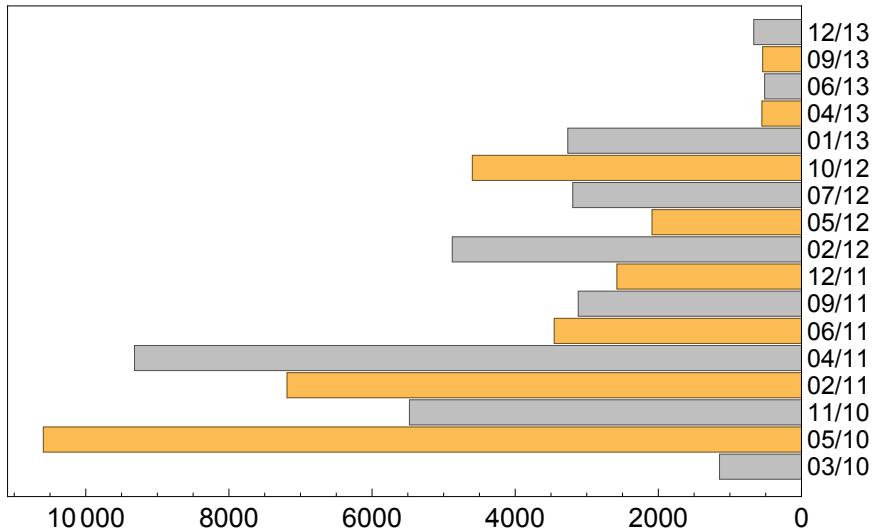
chartColors = {█, Lighter[Gray, .5]}
{█, █}

```

```

insetBars = BarChart[toBar[[Range[2, 18]]], ImageSize -> baseImageSize * 0.6,
Frame -> True, FrameTicks -> {{None, Automatic}, {Automatic, None}},
ChartLabels -> {Style[#, 13] & /@ gencodeAppearDateTicks[[Range[2, 18], 2]]},
BaseStyle -> {FontFamily -> "Arial", FontSize -> 14},
ChartStyle -> {chartColors[[2]], chartColors[[1]]},
PlotRangePadding -> {{500, 0}, {0.3, 0.5}}, BarOrigin -> Right]

```



```

toBar = Length /@ toBoxWhisker
{251810, 1145, 10593, 5478, 7188, 9319, 3455,
3119, 2579, 4879, 2088, 3196, 4599, 3265, 553, 515, 542, 667}

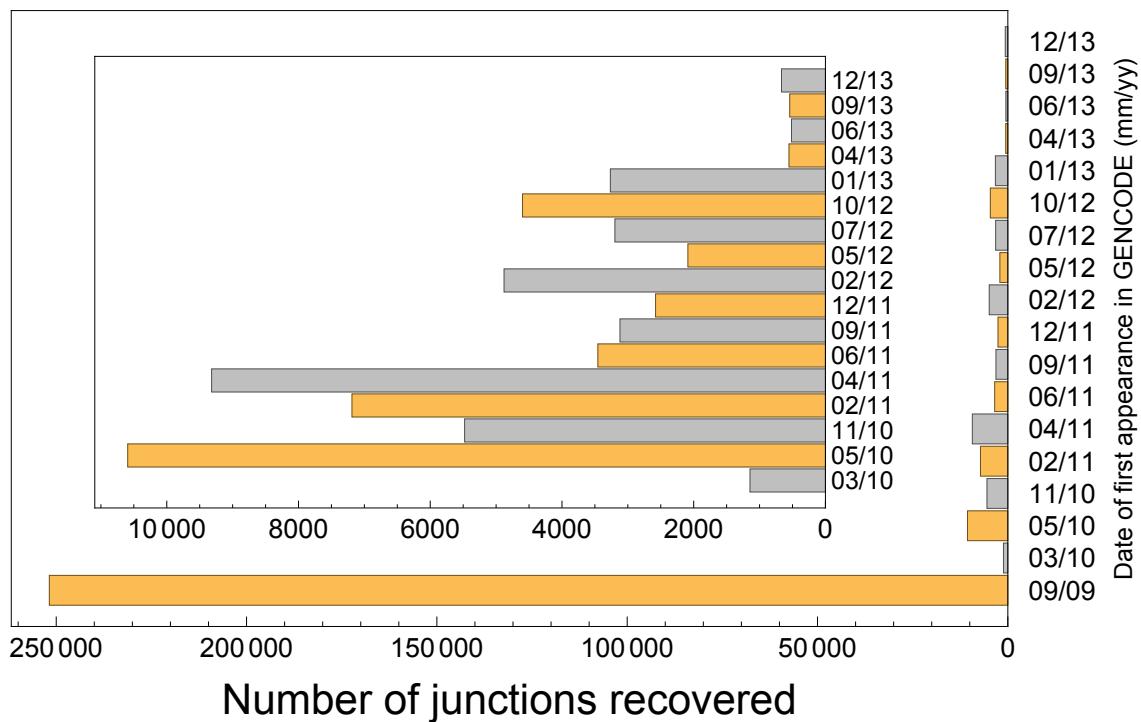
toBar[[1]] / Total[toBar] // N
0.799422

```

```

padding = {{10, 70}, {50, 0}};
barsWithInset = BarChart[toBar, ImageSize -> baseImageSize *.8,
  Frame -> True, FrameTicks -> {{None, Automatic}, {Automatic, None}},
  PlotRangePadding -> {{10000, 0}, {.3, .5}},
  ChartLabels -> {StringJoin[" ", #] & /@ gencodeAppearDateTicks[[All, 2]]},
  BaseStyle -> {FontFamily -> "Arial", FontSize -> 14},
  FrameLabel -> {{None, Style["Date of first appearance in GENCODE (mm/yy)", 13]}, {Style["Number of junctions recovered", 22], None}},
  ChartStyle -> chartColors, ImagePadding -> padding, BarOrigin -> Right,
  Epilog -> Inset[insetBars, {-135000, 10}, Automatic, 210000]]

```



```
magFlipped = Import["magflipped.png"]
```



```
origDateTicks =
  ({#,DateString[daysToDate[#], dateFormat]} & /@ Range[0, 2070, 329.3])
{{0., 02/27/09}, {329.3, 01/22/10}, {658.6, 12/17/10}, {987.9, 11/11/11},
 {1317.2, 10/06/12}, {1646.5, 08/31/13}, {1975.8, 07/26/14}}


newDateTicks = {{0.^, "02/27/09"}, {329.3^, "01/22/10"},
  {658.6^, "12/17/10"}, {987.900000000001^, "11/11/11"}, {1317.2^, "10/06/12"},
  {1646.5^, "08/31/13"}, {1975.800000000002^, "07/26/14"}}
{{0., 02/27/09}, {329.3, 01/22/10}, {658.6, 12/17/10},
 {987.9, 11/11/11}, {1317.2, 10/06/12}, {1646.5, 08/31/13}, {1975.8, 07/26/14}}


sampleCountsInFirstGencode =
  #[[2]] & /@ Select[junctionsEvidenceVsDatesGeq20, #[[5]] == 1 &];
sampleCountsInOtherGencodes = #[[2]] & /@ Select[junctionsEvidenceVsDatesGeq20,
  #[[5]] == 0 && Length[Position[#[[Range[5, 22]]], 1]] != 0 &];

anotherLabelForm [x_, y___] :=
  Text[Style[x, FontFamily -> "Arial", FontSize -> 14, Bold, TextAlignment -> Left], y]
```

```
daysToDate[329]
 Fri 22 Jan 2010

proportionOfTotalAt329 = ToString[
  NumberForm[N[Select[accumulatedJunctionsGeq[20], #[[1]] <= 329 &][[-1]][[2]]] /
    accumulatedJunctionsGeq[20][[-1]][[2]] * 100, 3], DigitBlock -> 3]
20.3

totBound = Select[accumulatedJunctionsGeq[20], #[[1]] <= 329 &][[-1]][[2]]
651 644

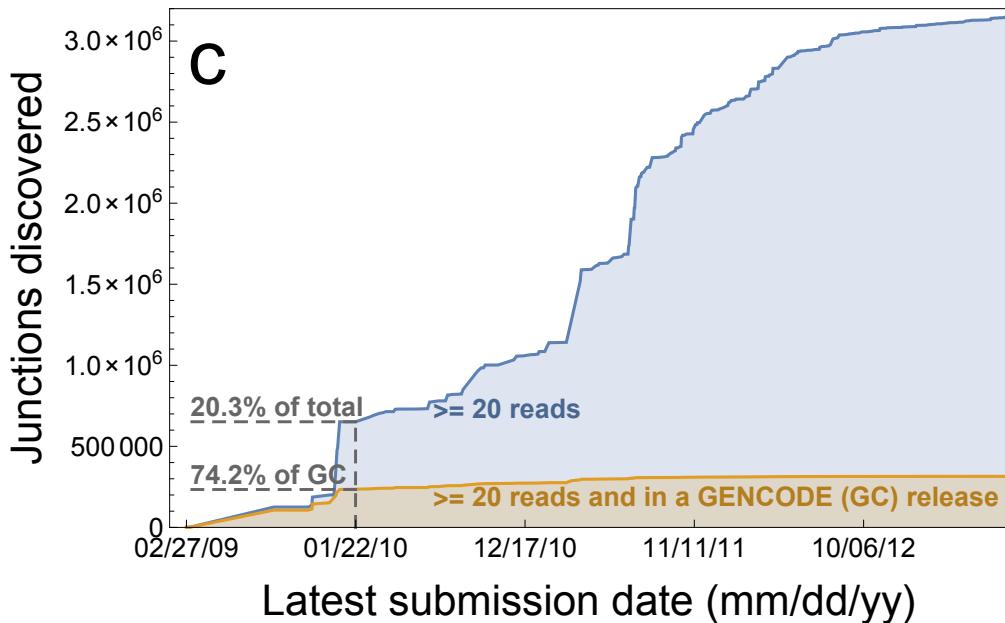
proportionOfAnnotatedAt329 =
  ToString[NumberForm[N[Select[accumulatedAnnotated, #[[1]] <= 329 &][[-1]][[2]]] /
    accumulatedAnnotated[[-1]][[2]] * 100, 3], DigitBlock -> 3]
74.2

annBound = Select[accumulatedAnnotated, #[[1]] <= 329 &][[-1]][[2]]
233 834
```

```

annJunctionsPlot = ListPlot[{accumulatedJunctionsGeg[20], accumulatedAnnotated},
  Joined → True, Filling → Axis, Frame → True,
  FrameTicks → {{Automatic, None}, {origDateTicks, None}},
  BaseStyle → {FontFamily → "Arial", FontSize → 14}, FrameLabel →
  {Style["Latest submission date (mm/dd/yy)", 22, TextAlignment → Left],
   Style["Junctions discovered", 22]}, ImageSize → baseImageSize * 0.7,
  PlotRange → {{Automatic, 1600}, {0, 3.2 * 10^6}}];
annJunctionsComplete = Show[annJunctionsPlot,
  Graphics[{Darker[mathematicaColors[[1]]], 0.2}, anotherLabelForm[">= 20 reads",
    {480, 735000}, {-1, 0}], Darker[mathematicaColors[[2]]], 0.2},
  anotherLabelForm[">= 20 reads and in a GENCODE (GC) release", {480, 187000},
    {-1, 0}], Directive[Thickness[0.0035], Dashing[0.013], labelColor],
  Line[{{329, 0}, {329, totBound}}], Line[{{329, annBound}, {0, annBound}}],
  anotherLabelForm[ToString[proportionOfAnnotatedAt329] <> "% of GC",
    {8, annBound + 90000}, {-1, 0}], Line[{{329, totBound}, {0, totBound}}],
  anotherLabelForm[ToString[proportionOfTotalAt329] <> "% of total",
    {8, totBound + 90000}, {-1, 0}]], Graphics[
  {Black, Text[Style["c", FontFamily → "Arial", FontSize → 40], {40, 2.9 * 10^6}]}]]

```



```

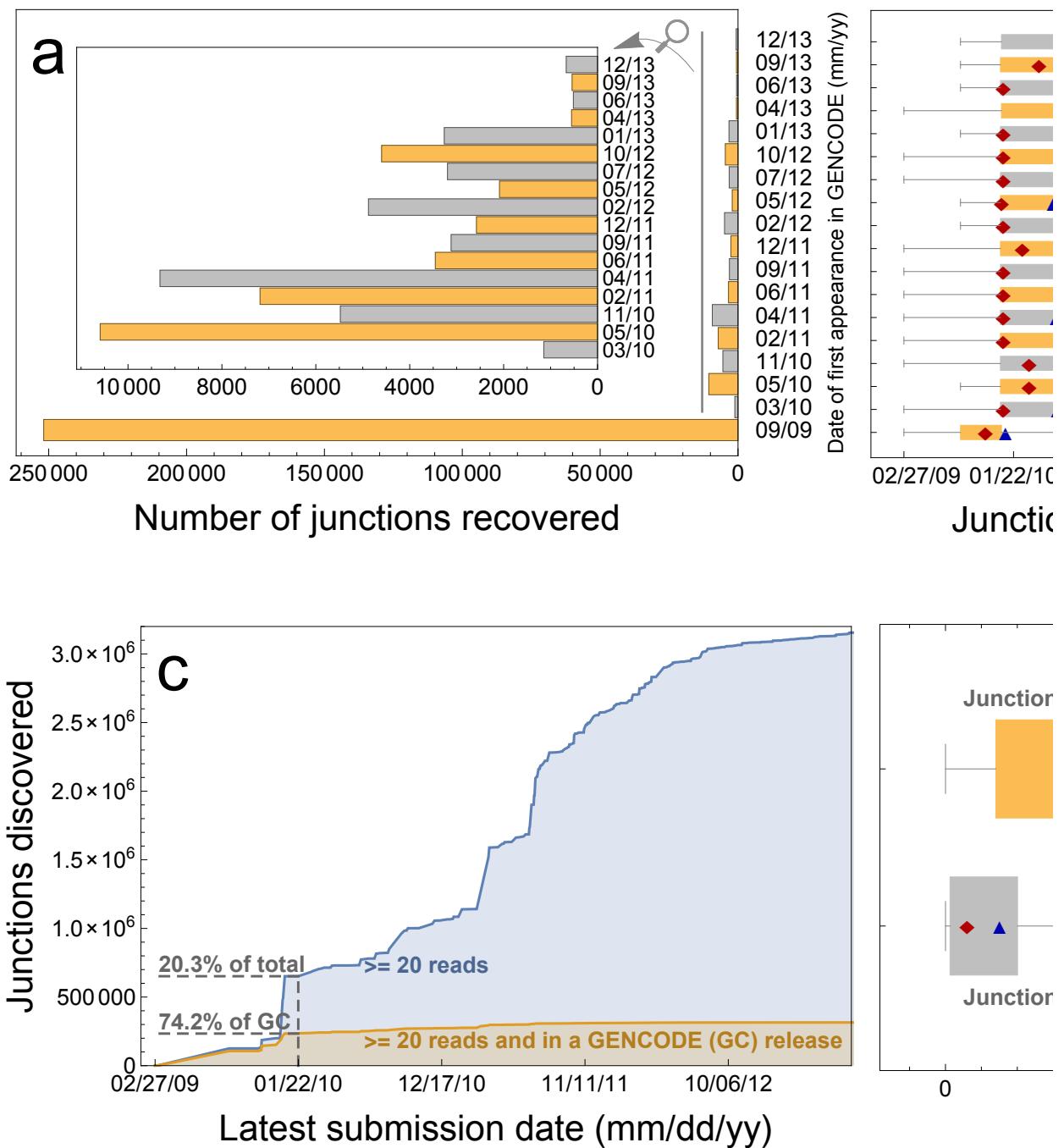
altLabelForm [x_, y___] := Text[Style[x, FontFamily → "Arial",
  FontSize → Scaled[.02], Bold, TextAlignment → Left], y]

```

```

sampleBoxPlot =
Show[BoxWhiskerChart[{sampleCountsInOtherGencodes, sampleCountsInFirstGencode},
  {"MeanMarker", "▲", Darker[Blue, 0.3]},
  {"MedianMarker", "◆", Darker[Red, 0.3]}}, ImageSize → baseImageSize * .613,
ChartStyle → {chartColors[[2]], chartColors[[1]]},
BarOrigin → Left, BarSpacing → Medium, Frame → True,
FrameLabel → {Style["Sample count", 22], None},
BaseStyle → {FontFamily → "Arial", FontSize → 14}],
Graphics[{Darker[Gray, 0.2], anotherLabelForm[
  "Junctions first appearing in first GENCODE release", {500, 2.45}, {-1, 0}],
anotherLabelForm["Junctions first appearing in other GENCODE releases",
{500, .55}, {-1, 0}]}], Graphics[
{Black, Text[Style["d", FontFamily → "Arial", FontSize → 40], {17 000, 2.65}]}]];
leftOff = -145 000; upOff = 90 000; suppevleft =
Show[barsWithInset, Graphics[
{Black, Text[Style["a", FontFamily → "Arial", FontSize → 40], {-250 000, 17.5}],
Gray, Arrow[BezierCurve[{{-16 000, 16.6}, {-30 000, 18.8}, {-45 000, 17.6}}]],
Thickness[.003], Line[{{-13 000, 18.3}, {-13 000, 1.8}}]}],
Graphics[{Opacity[0.5], Inset[magFlipped, {-30 000, 17.4}, {0, 0}, 12 000]}],
ImageSize → baseImageSize * 0.7];
otherpadding = {{0, 10}, {50, 0}};
suppevright =
Show[BoxWhiskerChart[tоБoxWhisker, {"MeanMarker", "▲", Darker[Blue, 0.3]},
 {"MedianMarker", "◆", Darker[Red, 0.3]}}, ImageSize → baseImageSize * .613,
ChartStyle → chartColors, BarOrigin → Left, BarSpacing → Medium, Frame → True,
PlotRange → {Automatic, Automatic}, PlotRangePadding → {{0.1, 0.9}, {0.3, 0.5}},
ImagePadding → otherpadding, FrameTicks → {{None, None}, {newDateTicks, None}},
BaseStyle → {FontFamily → "Arial", FontSize → 14},
FrameLabel → {Style["Junction discovery date (mm/dd/yy)", 22], None}], Graphics[
{Black, Text[Style["b", FontFamily → "Arial", FontSize → 40], {1985, 17.8}]}]];
suppeval = Grid[{{Grid[{{suppevleft, suppevright}}]}},
{Grid[{{annJunctionsComplete, sampleBoxPlot}}]}]

```



```
Export["ev.pdf", suppeval1]
```

```
ev.pdf
```

Assess strength of correlation between discovery date and Gencode date. Even rank correlation is small.

```
SpearmanRankTest[discoveryDaysToGencodeDays[[All, 1]],
discoveryDaysToGencodeDays[[All, 2]], "TestDataTable"] // N
```

| Statistic | P-Value |
|---------------|------------------------------|
| Spearman Rank | $0.356502 \times 10^{-9300}$ |

Exclude 2/28/09; relationship between it and the rest may be the dominant effect.

```
daysToDate[186]
```

 Tue 1 Sep 2009

```
discoveryDaysToGencodeDaysNoFirst =
Select[discoveryDaysToGencodeDays, #[[2]] != 186 &];

SpearmanRankTest[discoveryDaysToGencodeDaysNoFirst[[All, 1]],
discoveryDaysToGencodeDaysNoFirst[[All, 2]], "TestDataTable"] // N
```

| Statistic | P-Value |
|---------------|------------|
| Spearman Rank | -0.0148139 |
| | 0.00019633 |

... and this is true. Now let's load and plot PCs.

```
pcData =
Import["!cat pcs_unannotated_with_pd.tsv | tr -s '[:blank:]' '\t'", "TSV"];

pcHeader = pcData[[1]]
{PC1, PC2, PC3, PC4, PC5, brain, lcl, blood, seqc_abrf, geuvadis, mixture}

pcData = pcData[[Range[2, Length[pcData]]]];

In[380]:= brain = Select[pcData, #[[7]] == 1 &]; lcl = Select[pcData, #[[8]] == 1 &];
blood = Select[pcData, #[[9]] == 1 &]; oneToZero = Select[pcData, #[[-1]] == "1:0" &];
threeToOne = Select[pcData, #[[-1]] == "3:1" &];
oneToThree = Select[pcData, #[[-1]] == "1:3" &];
zeroToOne = Select[pcData, #[[-1]] == "0:1" &];
others = Select[pcData, #[[7]] != 1 && #[[8]] != 1 && #[[9]] != 1 && #[[-1]] == "NA" &];
geuvadis = Select[pcData, #[[-2]] == 1 &];
abrf = Select[pcData, #[[-3]] == 2 &]; seqc = Select[pcData, #[[-3]] == 1 &];

mathematicaColors
{
```

```
In[435]:= arrowLabelForm[x_, y___] :=
  Text[Style[x, FontFamily -> "Arial", FontSize -> Scaled[.03],
    Bold, TextAlignment -> Left], y]; labelColor = Darker[Gray, 0.2];
pcaPlot = Show[ListPlot[Transpose[{#[[All, 2]], #[[All, 3]]}]] & /@ {others, blood,
  lcl, brain, oneToZero, oneToThree, threeToOne, zeroToOne}, Frame -> True,
  ImageSize -> baseImageSize, PlotRange -> {{-.008, .015}, {-0.02, .023}}, Axes -> None,
  FrameLabel -> {Style["PC1", 22, TextAlignment -> Left], Style["PC2", 22]},
  BaseStyle -> {FontFamily -> "Arial", FontSize -> 14},
  PlotStyle -> {{PointSize[.006], Gray}, {PointSize[.006], mathematicaColors[[4]]},
    {PointSize[.006], Magenta}, {PointSize[.006], Cyan},
    {PointSize[.006], mathematicaColors[[15]]}, PointSize[.006],
    PointSize[.006], PointSize[.006]}], Graphics[{Opacity[0.25], Polygon[
  {{-.005, -.002}, {-0.005, .003}, {-0.001, .0073}, {0.0005, -0.0015}}], Polygon[
  {{.0075, .004}, {.0103, .012}, {.0048, .016}, {0.01, .026}, {0.0147, .016}, {0.015
  , 0.005}}], Polygon[{{.0054, -.015}, {0.005, -.008},
  {0.0095, -.013}, {0.01, -.019}}], Opacity[1],
  labelColor, Arrow[{{-.006, .008}, {-0.0035, .005}}]],
  arrowLabelForm["ABRF", {-0.0069, .0092}, {-1, 0}],
  Arrow[BezierCurve[{{.002, .016}, {0.004, .01}, {0.007, .0142}}]],
  arrowLabelForm["SEQC", {0.001, .017}, {-1, 0}],
  Arrow[BezierCurve[{{0, -.016}, {0.002, -.013}, {0.004, -.024}, {0.0063, -.016}}]],
  arrowLabelForm["GEU", {-0.001, -.0172}, {-1, 0}]]]
```

