```
Needs["ErrorBarPlots`"]
Needs["PlotLegends`"]

conv = 1.62359; (* multiply centroid_energy, part #76,
by this to convert to kT [all other energies are in kt] *)
```

# IMPORT DATA:

```
d50 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_50_MMA.
       xls", "XLS"], 1];
d100 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_100_MMA.
       xls", "XLS"], 1];
d200 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_200_MMA.
       xls", "XLS"], 1];
d400 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_400_MMA.
       xls", "XLS"], 1];
d800 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_800_MMA.
       xls", "XLS"], 1];
d1500 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_1500_MMA.
       xls", "XLS"], 1];
d2000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_2000_MMA.
       xls", "XLS"], 1];
d2500 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_2500_MMA.
       xls", "XLS"], 1];
d3000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_3000_MMA.
       xls", "XLS"], 1];
d4000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_4000_MMA.
       xls", "XLS"], 1];
d5000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_5000_MMA.
       xls", "XLS"], 1];
d6000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_6000_MMA.
       xls", "XLS"], 1];
d7000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_7000_MMA.
       xls", "XLS"], 1];
d8000 = Flatten[Import[
     "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Rand_Viral_8000_MMA.
       xls", "XLS"], 1];
```

```
{Length[d50], Length[d100], Length[d200], Length[d400], Length[d800],
 Length[d1500], Length[d2000], Length[d2500], Length[d3000], Length[d4000],
 Length[d5000], Length[d6000], Length[d7000], Length[d8000]}
```

```
{2000, 2000, 2000, 2000, 1000, 1000, 500, 500, 500, 500, 300, 300, 300, 1000}
```

```
bromo3 = Flatten[
   Import["/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Bromovirus
      _Cucomovirus_RNA3_MMA.xls", "XLS"], 1];
bromo2 = Flatten[
   Import["/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Bromovirus
      _Cucomovirus_RNA2_MMA.xls", "XLS"], 1];
bromo1 = Flatten[
   Import["/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Bromovirus
      _Cucomovirus_RNA1_MMA.xls", "XLS"], 1];
hepadna = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Hepadnaviridae_MMA.
      xls", "XLS"], 1];
levi = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Levivirus_MMA.xls",
    "XLS"], 1];
sobemo = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Sobemovirus_MMA.xls"
    , "XLS"], 1];
luteo = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Luteoviridae_MMA.xls
      ", "XLS"], 1];
tymo = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Tymovirus_MMA.xls",
    "XLS"], 1];
tobamo = Flatten[Import[
    "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_ALL/Vienna_Tobamovirus_MMA.xls"
    , "XLS"], 1];
astro = Import[
   "/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_Astroviridae/Vienna_Astroviridae
     _MMA.csv", "CSV"];
calic = Import["/Users/Aron/DOCUMENTS/FOLDING_RESULTS/Vienna_Caliciviridae/
     Vienna_Caliciviridae_MMA.csv", "CSV"];
```

```
{Length[bromo3], Length[bromo2], Length[bromo1],
 Length[hepadna], Length[levi], Length[sobemo], Length[luteo],
 Length[tymo], Length[tobamo], Length[astro], Length[calic]}
```

```
{8, 8, 8, 10, 9, 9, 17, 9, 22, 6, 18}
```

# FORMAT DATA :

```
t50 = Table[Length[d50[[i]]], {i, 1, Length[d50]}];
t100 = Table[Length[d100[[i]]], {i, 1, Length[d100]}];
t200 = Table[Length[d200[[i]]], {i, 1, Length[d200]}];
t400 = Table[Length[d400[[i]]], {i, 1, Length[d400]}];
t800 = Table[Length[d800[[i]]], {i, 1, Length[d800]}];
t1500 = Table[Length[d1500[[i]]], {i, 1, Length[d1500]}];
t2000 = Table[Length[d2000[[i]]], {i, 1, Length[d2000]}];
t2500 = Table[Length[d2500[[i]]], {i, 1, Length[d2500]}];
t3000 = Table[Length[d3000[[i]]], {i, 1, Length[d3000]}];
t4000 = Table[Length[d4000[[i]]], {i, 1, Length[d4000]}];
t5000 = Table[Length[d5000[[i]]], {i, 1, Length[d5000]}];
t6000 = Table[Length[d6000[[i]]], {i, 1, Length[d6000]}];
t7000 = Table[Length[d7000[[i]]], {i, 1, Length[d7000]}];
t8000 = Table[Length[d8000[[i]]], {i, 1, Length[d8000]}];
tbromo3 = Table[Length[bromo3[[i]]], {i, 1, Length[bromo3]}];
tbromo2 = Table[Length[bromo2[[i]]], {i, 1, Length[bromo2]}];
tbromo1 = Table[Length[bromo1[[i]]], {i, 1, Length[bromo1]}];
tlevi = Table[Length[levi[[i]]], {i, 1, Length[levi]}];
tsobemo = Table[Length[sobemo[[i]]], {i, 1, Length[sobemo]}];
tluteo = Table[Length[luteo[[i]]], {i, 1, Length[luteo]}];
ttymo = Table[Length[tymo[[i]]], {i, 1, Length[tymo]}];
tastro = Table[Length[astro[[i]]], {i, 1, Length[astro]}];
ttobamo = Table[Length[tobamo[[i]]], {i, 1, Length[tobamo]}];
tcalic = Table[Length[calic[[i]]], {i, 1, Length[calic]}];
```

```
{{Min[t50], Max[t50]}, {Min[t100], Max[t100]}, {Min[t200], Max[t200]},
 {Min[t400], Max[t400]}, {Min[t800], Max[t800]}, {Min[t1500], Max[t1500]},
 {Min[t2000], Max[t2000]}, {Min[t2500], Max[t2500]},
 {Min[t3000], Max[t3000]}, {Min[t4000], Max[t4000]}, {Min[t5000], Max[t5000]},
 {Min[t6000], Max[t6000]}, {Min[t7000], Max[t7000]}, {Min[t8000], Max[t8000]}}
```

```
{{77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77},
 {77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77}, {77, 77}}
```

# CREATE GLYPHS FOR PLOTS :

```
plotmarkers = {
    {Graphics[Disk[]], .035},
    {Graphics[
      {Thickness[0.05], Black, GraphicsComplex[{{-Sqrt[2] / 2, 0}, {0, -Sqrt[2] / 2},
          {Sqrt[2] / 2, 0}, {0, Sqrt[2] / 2}}, Line[{1, 2, 3, 4, 1}]]}], 0.035 * 1.15},
    {Graphics[Polygon[{{0, 0}, {1, 0}, {1 / 2, Sqrt[3] / 2}}]], 0.035},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[
        {{0, 0}, {0, 1}, {1, 1}, {1, 0}}, Line[{1, 2, 3, 4, 1}]]}], 0.035 * .9},
    {Graphics[{Thickness[0.05], Black, Circle[]}], .035},
    {Graphics[Rectangle[]], .035 * .9},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[
        {{0, 0}, {0, -1}, {0, 1}, {-1, 0}, {1, 0}}, Line[{2, 3, 1, 5, 4}]]}], 0.035},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[
        {{0, 0}, {1, 0}, {1 / 2, Sqrt[3] / 2}}, Line[{1, 2, 3, 1}]]}], 0.035},
    {Graphics[Polygon[{{0, 0}, {1, 0}, {1 / 2, -Sqrt[3] / 2}}]], 0.035}};
size = 8;
lm = {
    {Graphics[Disk[], ImageSize → size]},
    {Graphics[{Thickness[0.05], Black,
       GraphicsComplex[{{-Sqrt[2] / 2, 0}, {0, -Sqrt[2] / 2}, {Sqrt[2] / 2, 0},
         {0, Sqrt[2] / 2}}, Line[{1, 2, 3, 4, 1}]]}, ImageSize → size * 1.15]},
    {Graphics[Polygon[{{0, 0}, {1, 0}, {1 / 2, Sqrt[3] / 2}}], ImageSize → size]},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[{{0, 0}, {0, 1},
         {1, 1}, {1, 0}}, Line[{1, 2, 3, 4, 1}]]}, ImageSize → size * .9]},
    {Graphics[{Thickness[0.05], Black, Circle[]}, ImageSize → size]},
    {Graphics[Rectangle[], ImageSize -> size * .9]},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[{{0, 0}, {0, -1}, {0, 1},
         {-1, 0}, {1, 0}}, Line[{2, 3, 1, 5, 4}]]}, ImageSize → size]},
    {Graphics[{Thickness[0.05], Black, GraphicsComplex[
        {{0, 0}, {1, 0}, {1 / 2, Sqrt[3] / 2}}, Line[{1, 2, 3, 1}]]}, ImageSize → size]},
    {Graphics[Polygon[{{0, 0}, {1, 0}, {1 / 2, -Sqrt[3] / 2}}], ImageSize → size]}};
```

# Plot ⟨MLD⟩ vs. Sequence Length:

```
string1 = "⟨MLD⟩"; string2 = "Sequence Length"; j = 14;
ATT = {{2500, Mean[Transpose[d2500][[j]]]},
  {3000, Mean[Transpose[d3000][[j]]]}, {4000, Mean[Transpose[d4000][[j]]]},
  {5000, Mean[Transpose[d5000][[j]]]}, {6000, Mean[Transpose[d6000][[j]]]},
  {7000, Mean[Transpose[d7000][[j]]]}, {8000, Mean[Transpose[d8000][[j]]]}}
loglogATT = Map[{Log[#[[1]]], Log[#[[2]]]} &, ATT];
loglogerrATT = {{{Log[2500], Log[Mean[Transpose[d2500][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d2500][[j]]] / Mean[Transpose[d2500][[j]]]]},
   {{Log[3000], Log[Mean[Transpose[d3000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d3000][[j]]] / Mean[Transpose[d3000][[j]]]]},
   {{Log[4000], Log[Mean[Transpose[d4000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d4000][[j]]] / Mean[Transpose[d4000][[j]]]]},
   {{Log[5000], Log[Mean[Transpose[d5000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d5000][[j]]] / Mean[Transpose[d5000][[j]]]]},
   {{Log[6000], Log[Mean[Transpose[d6000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d6000][[j]]] / Mean[Transpose[d6000][[j]]]]},
   {{Log[7000], Log[Mean[Transpose[d7000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d7000][[j]]] / Mean[Transpose[d7000][[j]]]]},
   {{Log[8000], Log[Mean[Transpose[d8000][[j]]]]}, ErrorBar[
      StandardDeviation[Transpose[d8000][[j]]] / Mean[Transpose[d8000][[j]]]]}};
VlogATT = Log[{Map[{#[[2]], #[[j]]} &, bromo2], Map[{#[[2]], #[[j]]} &, bromo1],
    Map[{#[[2]], #[[j]]} &, levi], Map[{#[[2]], #[[j]]} &, sobemo],
    Map[{#[[2]], #[[j]]} &, luteo], Map[{#[[2]], #[[j]]} &, tymo],
    Map[{#[[2]], #[[j]]} &, tobamo], Map[{#[[2]], #[[j]]} &, astro],
    Map[{#[[2]], #[[j]]} &, calic]}];
lfitATT = Fit[loglogATT, {1, Nbases}, Nbases]
perrATT = ErrorListPlot[loglogerrATT,
   PlotStyle → {Black, Thickness[.002], PointSize[Medium]}];
pfitATT = Plot[lfitATT, {Nbases, Log[2400], Log[8800]},
   PlotStyle -> {Black, Thickness[.002]}];
pVATT = ListPlot[VlogATT, PlotMarkers → plotmarkers];
reg1 = LinearModelFit[loglogATT, Nbases, Nbases];
reg1["RSquared"]
tickmarksX = {{Log[2500], 2500}, {Log[3000], 3000}, {Log[4000], 4000},
   {Log[5000], 5000}, {Log[6000], 6000}, {Log[7000], 7000}, {Log[8000], 8000}};
tickmarksY = {{Log[150], 150}, {Log[200], 200}, {Log[300], 300},
   {Log[400], 400}, {Log[500], 500}, {Log[600], 600}, {Log[700], 700}};
tickmarksXL = Transpose[tickmarksX][[1]];
tickmarksYL = Transpose[tickmarksY][[1]];
tickmarksXN = Table[{tickmarksXL[[i]], Null}, {i, 1, Length[tickmarksXL]}];
tickmarksYN = Table[{tickmarksYL[[i]], Null}, {i, 1, Length[tickmarksYL]}];
amldplot = Show[perrATT, pfitATT, pVATT,
  Frame → True,
  FrameTicks → {{tickmarksY, tickmarksYN}, {tickmarksX, tickmarksXN}},
  FrameTicksStyle → Directive[12],
  AxesOrigin → {Log[2400], Log[140]},
  PlotRange → {Log[140], Log[750]},
  FrameLabel → {"Sequence Length", string1},
  LabelStyle → {Directive[Black, 12, FontFamily → "Times"]},
  FrameStyle → {{Black, Thickness[0.002]}, {Black, Thickness[0.002]},
    {Black, Thickness[0.002]}, {Black, Thickness[0.002]}},
  ImageSize → 400, Epilog -> Inset[
```

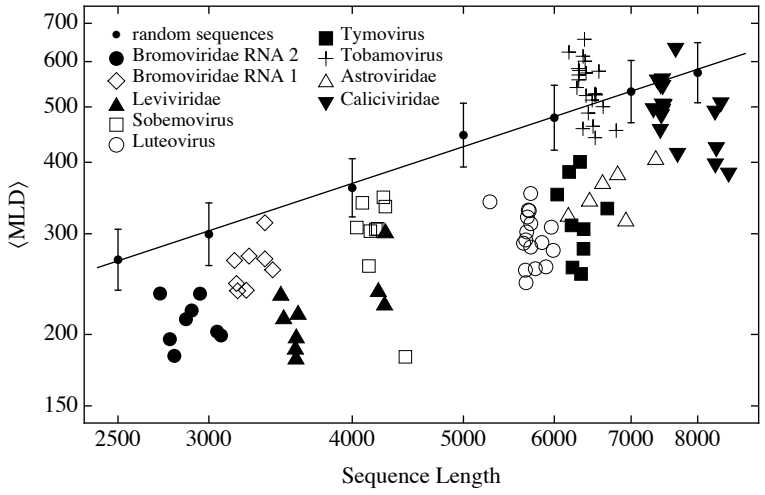
```
    Grid[{{"•", "random sequences", lm[[6, 1]], "Tymovirus"},
      {lm[[1, 1]], "Bromoviridae RNA 2",  lm[[7, 1]], "Tobamovirus"},
      {lm[[2, 1]], "Bromoviridae RNA 1", lm[[8, 1]], "Astroviridae"},
      {lm[[3, 1]], "Leviviridae", lm[[9, 1]], "Caliciviridae"},
  {lm[[4, 1]], "Sobemovirus", Null, Null}, {lm[[5, 1]], "Luteovirus", Null, Null}},
     Alignment → {{Center, Left, Center, Left}},
     ItemSize -> {Automatic, Automatic},
  ItemStyle -> {Directive[10, FontFamily → "Times"], None},
     Spacings -> {{0, .4, 1, .4}, .13}], Scaled[{0.285, 0.8}]]
 ]
```

```
Export["/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnas2.pdf", amldplot]
Export["/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnas2.eps", amldplot]
```

```
{{2500, 270.213}, {3000, 299.438}, {4000, 360.977},
 {5000, 446.295}, {6000, 478.654}, {7000, 531.756}, {8000, 573.783}}
```

```
0.405373+ 0.663254 Nbases
```

```
0.994205
```




```
/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnas2.pdf
```

```
/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnas2.eps
```

# Plot Log (Frequency MFE Structure) vs. Sequence Length:

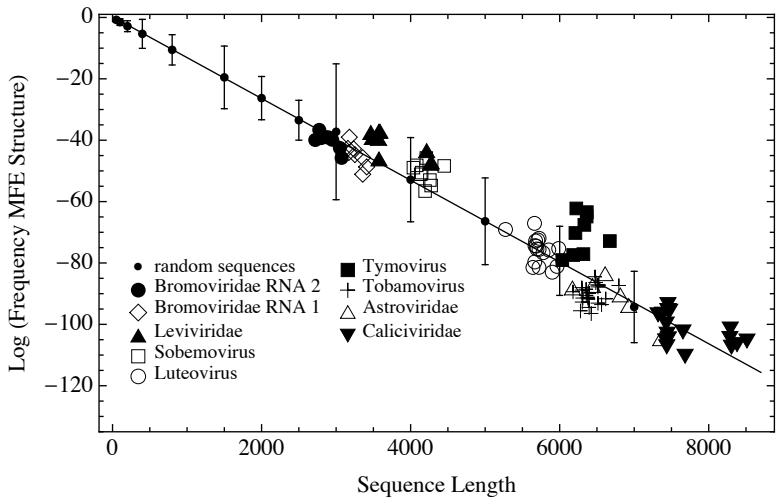
```
string1 = "Log (Frequency MFE Structure)"; string2 = "Sequence Length";
j = 11;
ATT = {{50, Mean[Transpose[d50][[j]]]},
   {100, Mean[Transpose[d100][[j]]]}, {200, Mean[Transpose[d200][[j]]]},
   {400, Mean[Transpose[d400][[j]]]}, {800, Mean[Transpose[d800][[j]]]},
   {1500, Mean[Transpose[d1500][[j]]]}, {2000, Mean[Transpose[d2000][[j]]]},
   {2500, Mean[Transpose[d2500][[j]]]}, {3000, Mean[Transpose[d3000][[j]]]},
   {4000, Mean[Transpose[d4000][[j]]]}, {5000, Mean[Transpose[d5000][[j]]]},
   {6000, Mean[Transpose[d6000][[j]]]}, {7000, Mean[Transpose[d7000][[j]]]}};
logATT = Map[{#[[1]], Log[10, #[[2]]]} &, ATT];
logerrATT = {{{50, Log[10, Mean[Transpose[d50][[j]]]]},
    ErrorBar[StandardDeviation[Transpose[d50][[j]]] / Mean[Transpose[d50][[j]]]]},
   {{100, Log[10, Mean[Transpose[d100][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d100][[j]]] / Mean[Transpose[d100][[j]]]]},
   {{200, Log[10, Mean[Transpose[d200][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d200][[j]]] / Mean[Transpose[d200][[j]]]]},
   {{400, Log[10, Mean[Transpose[d400][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d400][[j]]] / Mean[Transpose[d400][[j]]]]},
   {{800, Log[10, Mean[Transpose[d800][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d800][[j]]] / Mean[Transpose[d800][[j]]]]},
   {{1500, Log[10, Mean[Transpose[d1500][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d1500][[j]]] / Mean[Transpose[d1500][[j]]]]},
   {{2000, Log[10, Mean[Transpose[d2000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d2000][[j]]] / Mean[Transpose[d2000][[j]]]]},
   {{2500, Log[10, Mean[Transpose[d2500][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d2500][[j]]] / Mean[Transpose[d2500][[j]]]]},
   {{3000, Log[10, Mean[Transpose[d3000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d3000][[j]]] / Mean[Transpose[d3000][[j]]]]},
   {{4000, Log[10, Mean[Transpose[d4000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d4000][[j]]] / Mean[Transpose[d4000][[j]]]]},
   {{5000, Log[10, Mean[Transpose[d5000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d5000][[j]]] / Mean[Transpose[d5000][[j]]]]},
   {{6000, Log[10, Mean[Transpose[d6000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d6000][[j]]] / Mean[Transpose[d6000][[j]]]]},
   {{7000, Log[10, Mean[Transpose[d7000][[j]]]]}, ErrorBar[
     StandardDeviation[Transpose[d7000][[j]]] / Mean[Transpose[d7000][[j]]]]}};
VlogATT = {Map[{#[[2]], Log[10, #[[j]]]} &, bromo2],
   Map[{#[[2]], Log[10, #[[j]]]} &, bromo1], Map[{#[[2]], Log[10, #[[j]]]} &, levi],
   Map[{#[[2]], Log[10, #[[j]]]} &, sobemo], Map[{#[[2]], Log[10, #[[j]]]} &, luteo],
   Map[{#[[2]], Log[10, #[[j]]]} &, tymo], Map[{#[[2]], Log[10, #[[j]]]} &, tobamo],
   Map[{#[[2]], Log[10, #[[j]]]} &, astro], Map[{#[[2]], Log[10, #[[j]]]} &, calic]};
lfitATT = Fit[logATT, {1, Nbases}, Nbases]
ATTplot = ListPlot[logATT];
perrATT = ErrorListPlot[logerrATT,
   PlotStyle → {Black, Thickness[.002], PointSize[Medium]}];
pfitATT = Plot[lfitATT, {Nbases, 30, 8700}, PlotStyle -> {Black, Thickness[.002]}];
pVATT = ListPlot[VlogATT, PlotMarkers → plotmarkers];
reg1 = LinearModelFit[logATT, Nbases, Nbases];
reg1["RSquared"]
```

```
amldplot = Show[perrATT, pfitATT, pVATT,
  Frame → True,
  FrameTicks → Automatic
  (*{{tickmarksY,tickmarksYN},{tickmarksX,tickmarksXN}}*),
  FrameTicksStyle → Directive[12],
  AxesOrigin → {40, Log[10, 10^-135]},
  PlotRange → {Log[10, 10^-135], Log[10, 10]},
  FrameLabel → {string2, string1},
  LabelStyle → {Directive[Black, 12, FontFamily → "Times"]},
  FrameStyle → {{Black, Thickness[0.002]}, {Black, Thickness[0.002]},
    {Black, Thickness[0.002]}, {Black, Thickness[0.002]}},
  ImageSize → 400,
  Epilog -> Inset[Grid[{{"•", "random sequences", lm[[6, 1]], "Tymovirus"},
      {lm[[1, 1]], "Bromoviridae RNA 2", lm[[7, 1]], "Tobamovirus"},
      {lm[[2, 1]], "Bromoviridae RNA 1", lm[[8, 1]], "Astroviridae"},
      {lm[[3, 1]], "Leviviridae", lm[[9, 1]], "Caliciviridae"},
  {lm[[4, 1]], "Sobemovirus", Null, Null}, {lm[[5, 1]], "Luteovirus", Null, Null}},
     Alignment → {{Center, Left, Center, Left}},
     ItemSize -> {Automatic, Automatic},
 ItemStyle -> {Directive[10, FontFamily → "Times"], None},
     Spacings -> {{0, .4, 1, .4}, .13}], Scaled[{0.295, 0.265}]]
 ]
```

```
Export["/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnasS1.pdf", amldplot]
Export["/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnasS1.eps", amldplot]
```

```
0.232498 - 0.0133192 Nbases
```

```
0.999252
```




```
/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnasS1.pdf
```

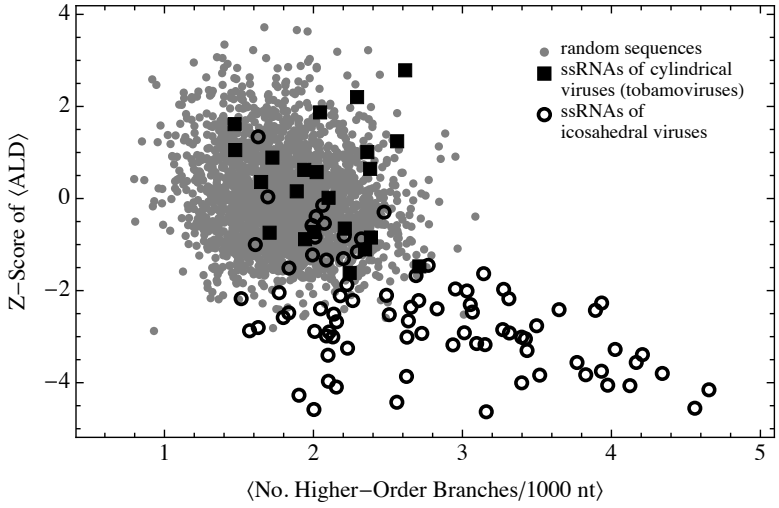
```
/Users/Aron/Documents/THESIS/SIZE/sizefigs/pnasS1.eps
```

# Plot ⟨No. Higher - Order Branches/1000 nt⟩ vs. Z - Score of ⟨ALD⟩:

```
plotmarkers2 = {
    {Graphics[{Gray, Disk[]}], 0.02},
    {Graphics[{Thickness[0.3], Black, Circle[]}], .035},
    {Graphics[Rectangle[]], .035 * .9}};
size = 8;
lm2 = {
    {Graphics[{Gray, Disk[]}, ImageSize → 5]},
    {Graphics[{Thickness[0.3], Black, Circle[]}, ImageSize → size]},
    {Graphics[Rectangle[], ImageSize -> size * .9]}};
AALD[nb_] := 0.5256973036711742` nb^0.67040861994172584736872977373423054814`15.954589770191005
stdevAALD[nb_] := 0.05214232411456724` nb^0.67437061753156646304319110640790313482`15.954589770191005
Vbranch4p = Flatten[{Map[{#[[2]], #[[21]], #[[44]]} &, bromo2],
     Map[{#[[2]], #[[21]], #[[44]]} &, bromo1],
     Map[{#[[2]], #[[21]], #[[44]]} &, levi], Map[{#[[2]], #[[21]], #[[44]]} &,
      sobemo], Map[{#[[2]], #[[21]], #[[44]]} &, luteo],
     Map[{#[[2]], #[[21]], #[[44]]} &, tymo], Map[{#[[2]], #[[21]], #[[48]]} &, astro],
     Map[{#[[2]], #[[21]], #[[48]]} &, calic]}, 1];
Tobamobranch4p = Map[{#[[2]], #[[21]], #[[44]]} &, tobamo];
Rbranch4p = Flatten[{Map[{#[[2]], #[[21]], #[[44]]} &, d2500],
     Map[{#[[2]], #[[21]], #[[44]]} &, d3000], Map[{#[[2]], #[[21]], #[[44]]} &,
      d4000], Map[{#[[2]], #[[21]], #[[44]]} &, d5000],
     Map[{#[[2]], #[[21]], #[[44]]} &, d6000], Map[{#[[2]], #[[21]], #[[44]]} &,
      d7000], Map[{#[[2]], #[[21]], #[[48]]} &, d8000]}, 1];
t3R = Table[{1000 * Rbranch4p[[i, 3]] / Rbranch4p[[i, 1]],
     (Rbranch4p[[i, 2]] - AALD[Rbranch4p[[i, 1]]]) / stdevAALD[Rbranch4p[[i, 1]]]},
    {i, 1, Length[Rbranch4p]}];
t3V = Table[{1000 * Vbranch4p[[i, 3]] / Vbranch4p[[i, 1]],
     (Vbranch4p[[i, 2]] - AALD[Vbranch4p[[i, 1]]]) / stdevAALD[Vbranch4p[[i, 1]]]},
    {i, 1, Length[Vbranch4p]}];
t3T = Table[{1000 * Tobamobranch4p[[i, 3]] / Tobamobranch4p[[i, 1]],
     (Tobamobranch4p[[i, 2]] - AALD[Tobamobranch4p[[i, 1]]]) /
      stdevAALD[Tobamobranch4p[[i, 1]]]}, {i, 1, Length[Tobamobranch4p]}];
ZscoreplotR = ListPlot[t3R, PlotMarkers → plotmarkers2[[1]]];
ZscoreplotV = ListPlot[t3V, PlotMarkers → plotmarkers2[[2]]];
ZscoreplotT = ListPlot[t3T, PlotMarkers → plotmarkers2[[3]]];
pfitZscore = Plot[lfitZscore, {Nbases, 0, 3.2}];
Zscore4 = Show[ZscoreplotR, ZscoreplotV, ZscoreplotT,
  PlotRange → {{.5, 5}, {-5, 4}},
  AxesOrigin → {.5, -5} ,
  FrameLabel → {"⟨No. Higher-Order Branches/1000 nt⟩", "Z-Score of ⟨ALD⟩"},
  Frame → True, LabelStyle → {Directive[Black, 12, FontFamily → "Times"]},
  FrameStyle → {{Black, Thickness[0.002]}, {Black, Thickness[0.002]},
    {Black, Thickness[0.002]}, {Black, Thickness[0.002]}},
  ImageSize → 400, Epilog -> Inset[Grid[{{lm2[[1, 1]], "random sequences"},
      {lm2[[3, 1]], "ssRNAs of cylindrical"}, {"", "viruses (tobamoviruses)"},
      {lm2[[2, 1]], "ssRNAs of"}, { "", "icosahedral viruses"}},
     Alignment → {{Center, Left}}, ItemSize -> {Automatic, Automatic},
 ItemStyle -> {Directive[10, FontFamily → "Times"], None},
     Spacings -> {{0, .5}, .13}], Scaled[{.81, 0.8}]]
```

```
]
```

```
Export["/Users/Aron/DOCUMENTS/THESIS/SIZE/sizefigs/Zbranch.pdf", Zscore4]
Export["/Users/Aron/DOCUMENTS/THESIS/MASTER_THESIS/sizefigs/Zbranch.pdf", Zscore4]
```




```
/Users/Aron/DOCUMENTS/THESIS/SIZE/sizefigs/Zbranch.pdf

/Users/Aron/DOCUMENTS/THESIS/MASTER_THESIS/sizefigs/Zbranch.pdf
```