NIBB GITC「BLAST 自由自在」
September 6-7, 2017@Okazaki

BLAST結果の処理法（２）可視化

Shuji Shigenobu /重信秀治

Aim
- BLASTの結果を可視化し、知識発見に結びつける。

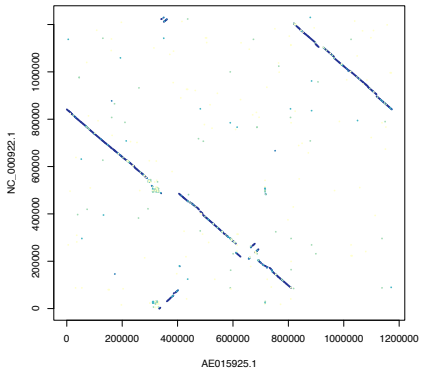
BLAST結果を表データのみから
理解するのは難しい

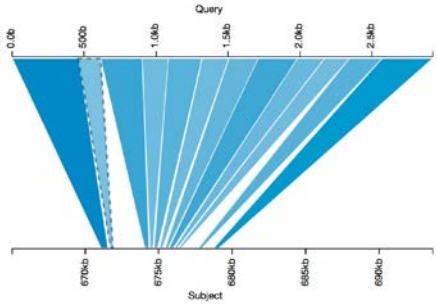
format 6

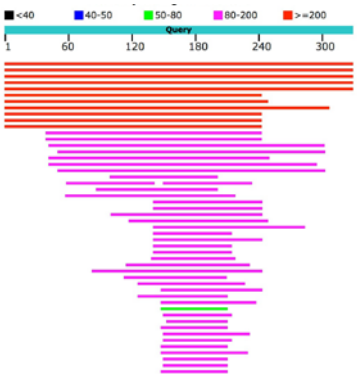
query	subject	%identity	align-len	mismatch	gap_open	q_start	q_end	s_start	s_end	evalue	bit-score
spo:SPAC212.11	sce:YMR190C	27.297	370	238	12	1169	1525	653	1004	7.08e-23	105
spo:SPAC212.11	sce:YDR021W	38.053	113	64	3	1417	1529	267	373	1.42e-14	75.9
spo:SPAC212.11	sce:YNL112W	25.076	331	225	12	1204	1525	148	464	1.06e-12	70.9
spo:SPAC212.11	sce:YBR237W	24.403	377	237	17	1190	1532	276	638	1.68e-12	70.5
spo:SPAC212.11	sce:YOR204W	28.505	214	128	8	1352	1559	339	533	2.80e-12	69.7
spo:SPAC212.11	sce:YOR046C	28.205	234	140	8	1310	1530	227	445	1.65e-11	66.6
spo:SPAC212.11	sce:YDR243C	23.512	336	227	10	1208	1524	216	540	7.40e-11	65.1
spo:SPAC212.11	sce:YGL078C	22.689	357	239	11	1188	1527	130	466	7.62e-11	64.7
spo:SPAC212.11	sce:YPL119C	27.619	210	129	8	1352	1556	351	542	2.84e-10	63.2
spo:SPAC212.11	sce:YGL064C	22.195	410	216	16	1215	1541	166	555	8.25e-10	61.6
spo:SPAC212.11	sce:YDL084W	24.424	217	156	5	1308	1524	201	409	7.84e-07	51.6
spo:SPAC212.11	sce:YHR169W	27.273	143	93	4	1418	1555	257	393	1.09e-06	51.2
spo:SPAC212.11	sce:YLR276C	26.667	210	120	10	1189	1377	38	234	2.85e-06	50.1
spo:SPAC212.11	sce:YLR276C	34.146	41	27	0	1484	1524	386	426	0.12	35.0
spo:SPAC212.11	sce:YLL008W	21.965	346	233	13	1195	1524	258	582	4.71e-06	49.3
spo:SPAC212.11	sce:YPL082C	24.074	108	81	1	1417	1524	1648	1754	0.002	40.8
spo:SPAC212.11	sce:YGL070C	37.838	37	23	0	1583	1619	52	88	2.3	29.3
spo:SPBPB10D8.05C	sce:YPL092W	30.730	397	224	9	10	356	11	406	3.04e-47	165
spo:SPBPB10D8.05C	sce:YGL195W	31.034	58	37	2	140	195	2014	2070	0.37	30.8
spo:SPBPB10D8.05C	sce:YDR283C	51.613	31	14	1	221	250	589	619	0.66	30.0
spo:SPBPB10D8.05C	sce:YBR028C	35.135	37	24	0	229	265	127	163	0.84	29.6
spo:SPBPB10D8.05C	sce:YPL027W	27.957	93	54	3	138	221	53	141	3.1	27.3
spo:SPBPB10D8.05C	sce:YDR161W	28.571	49	35	0	318	366	102	150	3.2	27.7
spo:SPBPB10D8.05C	sce:YER166W	27.451	51	35	1	50	98	1273	1323	7.0	26.9
spo:SPBPB10D8.05C	sce:YGR040W	23.256	86	55	3	214	299	3	77	7.8	26.6
spo:SPBPB10D8.05C	sce:YAR019C	44.444	18	10	0	235	252	30	47	9.5	26.2
spo:SPBC359.02	sce:YOR368W	29.885	87	56	2	53	138	7	89	0.88	29.3
spo:SPCC330.07C	sce:YHR197W	26.761	71	39	3	65	130	287	349	2.0	28.9
spo:SPCC330.07C	sce:YGR224W	23.881	67	50	1	181	247	152	217	5.1	27.7

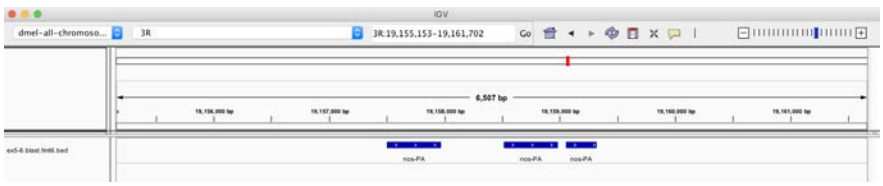
- BED file conversion => IGV
- Kablammo
- SequenceServer
- Dotplot

Visualize BLAST outputs









Visualize with IGV

- Strategy
  - BLAST output (format 6)
  - => convert to BED format
  - => visualize with IGV

IGVの詳細については、GITC RNA-seq 入門や、IGV wevsiteを参照
- http://software.broadinstitute.org/software/igv/
- https://github.com/nibb-gitc/gitc2018july-rnaseq/blob/master/textbook/3-IGV.pdf

Ex5-6

Let's try! (ex 5-6)

Ex1-1 のBLAST結果をBEDフォーマットに変換し、IGVで可視化しよう。
(mapping cDNA to a genome)
blastn の練習。キイロショウジョウバエのnos遺伝子のORFの塩基配列をblastnでゲノムにマッピングし、エクソン／イントロン構造を大まかに把握する。

1. BLAST search

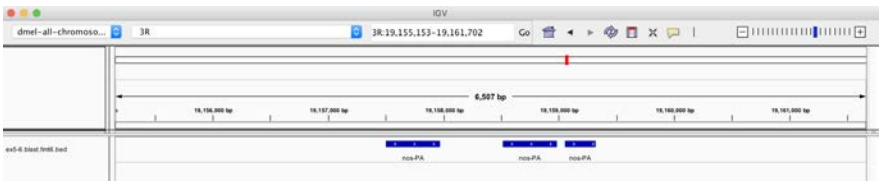
```
$blastn -query Dmel_nos-PA.nuc.fasta -db dmel-all-chromosome-r6.13.fasta -evalue 1.0e-20 -outfmt 6 > ex5-6.blast.fmt6.txt
```

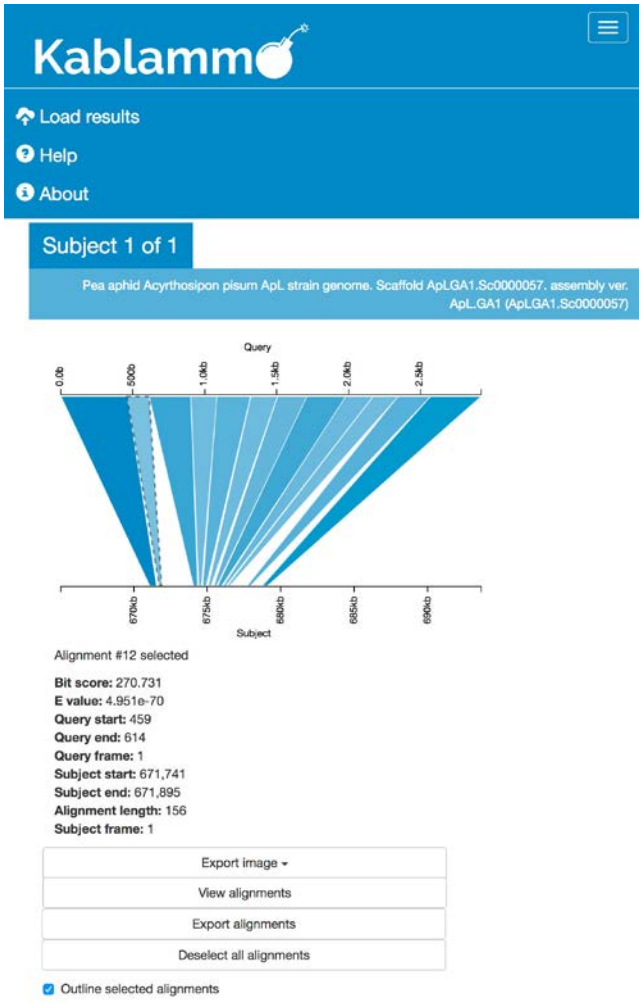
2. Convert blast table to BED format

```
$ruby blast6_to_bed.rb ex5-6.blast.fmt6.txt > ex5-6.blast.fmt6.bed
```

3. Visualize with IGV

```
3-1: Create genome
3-2: Load BED file
```





http://kablammo.wasmuthlab.org/

- Kablammo is a web-based BLAST visualizer
- Easy setup. No need to be installed by users.
- Just upload BLAST results in XML format

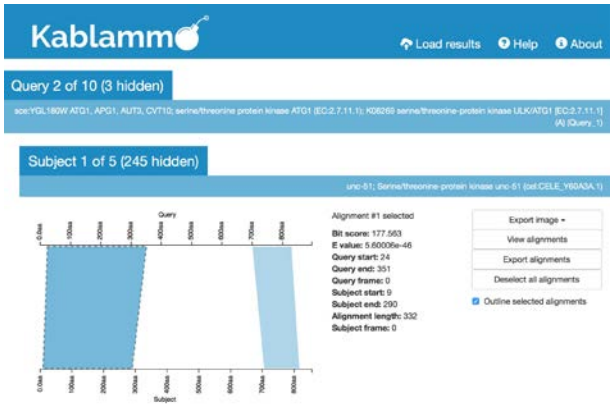
Ex5-4

Practice: ex5-4

ex1-2で解析した、BLASTP の結果をKablammo で可視化しよう。
XML format で出力し、それをKablammoにアップロード

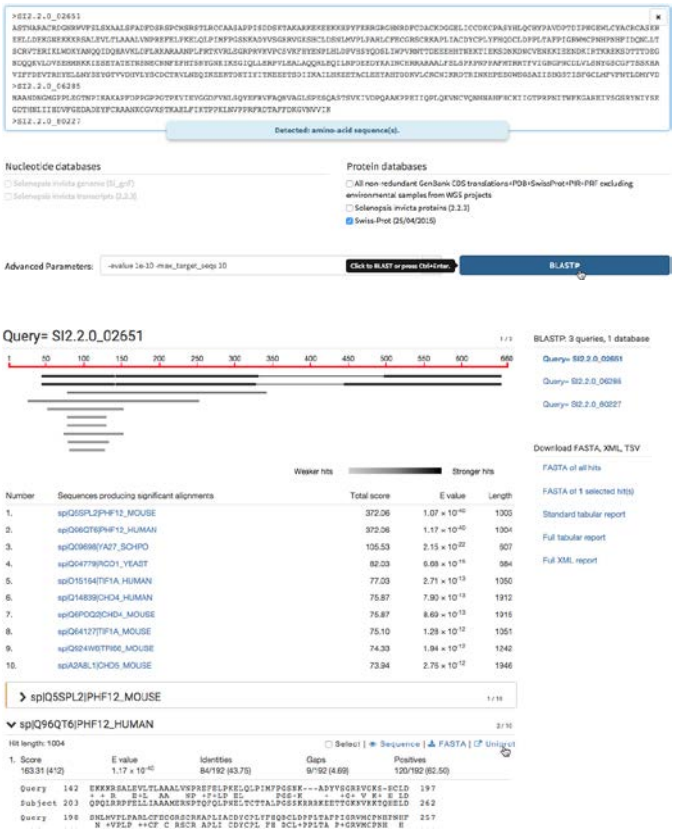
URL: http://kablammo.wasmuthlab.org/

```
blastp -query ScATGgenes.aa.fasta -db Cele.T00019.pep –outfmt 5 \
   > out.xml
```



SequenceServer

http://www.sequenceserver.com/



- SequenceServer is a web-based interface of BLAST+.
- A visualization function like Web version of NCBI BLAST was implemented recently.
- Easy to set up.

Ex5-5

Let's try

Sequenceserver をインストールし、線虫 *C. elegans* のタンパク質BLAST database を構築する。
C. elegans タンパク質のアミノ酸配列は、ex1-2 で使った, Cele.T00019.pep

1. Install sequenceserver

```
#今回使うMacにはインストール済みなので以下のコマンドは実行する必要はない
$sudo gem install sequenceserver
```

2. Set up blastdb

```
makeblastdb -in Cele.T00019.pep  -dbtype prot –parse_seqids
```

3. Start server

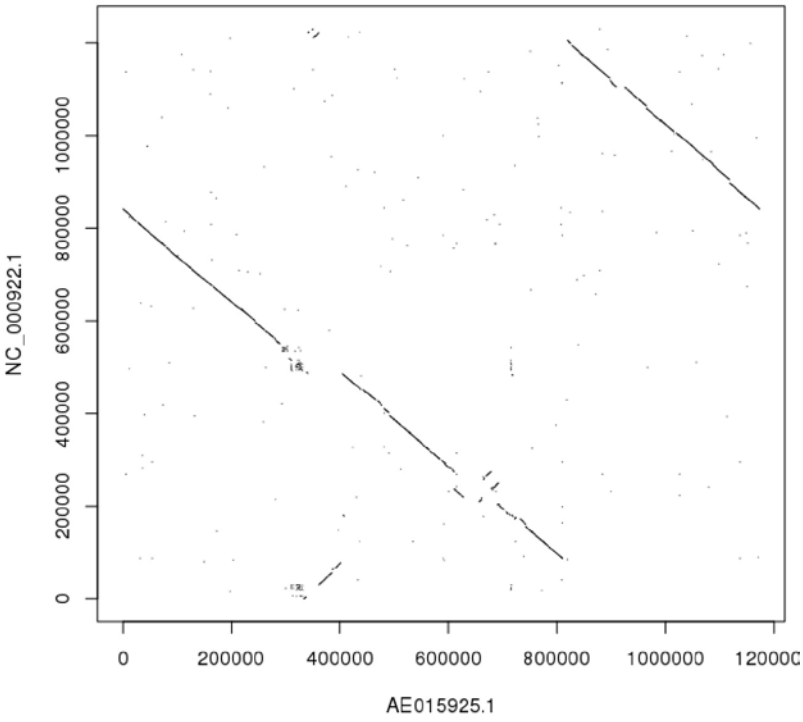
```
$sequencesever
```

4. Access sequencesever with web browser

```
(access http://localhost:4567)
```

5. BLAST search and automatic visualization

Dotplot



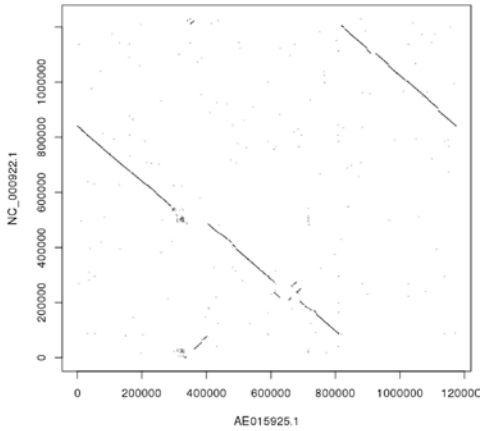
query	subject	%identity				q_start	q_end	s_start	s_end	evalue	bit-score
AE015925.1	NC_000922.1	79.515	14972	2951	49	793170	808087	104228	89319	0.0	13057
AE015925.1	NC_000922.1	77.062	14439	3149	68	107763	122130	730578	716232	0.0	10947
AE015925.1	NC_000922.1	70.626	23187	6113	286	874380	897222	1146890	1124058	0.0	10448
AE015925.1	NC_000922.1	76.018	12205	2796	72	1057699	1069843	968388	956255	0.0	8608
AE015925.1	NC_000922.1	94.369	5203	244	27	1020573	1025748	1005510	1000330	0.0	7984
AE015925.1	NC_000922.1	68.306	17729	5158	216	965915	983437	1057574	1040101	0.0	6082
AE015925.1	NC_000922.1	69.808	13550	3748	180	1029907	1043269	996470	983077	0.0	5490
…											

a short script

Table for R

```
AE015925.1      NC_000922.1
793170  104228
808087  89319
NA      NA
107763  730578
122130  716232
NA      NA
874380  1146890
897222  1124058
NA      NA
…
```

```
[R]
> dat <- read.delim("table.dat")
> plot(x, type="line")
```



Ex5-1

Let's try! (ex 5-1)

肺炎クラミジア Chlamydophila pneumoniae のCWL029株と、GPIC株のゲノム全長をBLASTNで比較し、dotplotを描画することによってゲノム構造の保存性を視覚化しなさい。

- Seq1: CpneCWL029.NC_000922.uc.genome.fa
- Seq2: CpneGPIC.AE015925.uc.genome.fa

```
blastn -task blastn -subject CpneCWL029.NC_000922.uc.genome.fa \
      -query CpneGPIC.AE015925.uc.genome.fa -outfmt 6 \
      > CpneCWL029.vs.CpneGPIC.blast.out.fmt6

ruby blast6_to_rplot1.rb CpneCWL029.vs.CpneGPIC.blast.out.fmt6 \
      > CpneCWL029.vs.CpneGPIC.blast.out.fmt6.mat
```

```
(R)
> dat <- read.delim("CpneCWL029.vs.CpneGPIC.blast.out.fmt6.mat")
> plot(dat, type="l", lwd=2)
```

Ex5-2

(参考: ex 5-2)

ex 5-1のdot plot描画を発展させ、bitscore よって色を変えてプロットするRスクリプトを自動生成する ruby scriptを用意した。試してみよう。

- Script: blast6_to_rplot2.rb

```
blastn -task blastn -subject CpneCWL029.NC_000922.uc.genome.fa \
      -query CpneGPIC.AE015925.uc.genome.fa -outfmt 6 \
      > CpneCWL029.vs.CpneGPIC.blast.out.fmt6

ruby blast6_to_rplot2.rb CpneCWL029.vs.CpneGPIC.blast.out.fmt6 \
      > CpneCWL029.vs.CpneGPIC.blast.out.fmt6.plot.R
```

```
(R)
> png("out.png", width=800, height=800)
> source("CpneCWL029.vs.CpneGPIC.blast.out.fmt6.plot.R")
> dev.off()
```