Local alignments & BLAST online and offline

Wen-Dar Lin

Bioinformatics core, IPMB
wdlin@gate.sinica.edu.tw

Preface

- When we are talking about *sequence similarity*, we usually are talking about *local alignments*.
- NCBI BLAST is one of the most famous local alignment programs where almost everyone has ever used it.
- The concept of BLAST algorithm is fundamental.
  - Knowing it would help us to understand many other modern alignment/mapping algorithms.

Preface

- This presentation is intended to provide information of
  - *theoretical background* of local alignments,
  - *underlying algorithm* of BLAST, and
  - *usages* of BLAST programs.
- Files: PowerPoints, walk-through logs, scripts, and example data
  - https://maccu.project.sinica.edu.tw/20250513/
    - would have some update by noon of 20250513

Disclaimer

- This presentation is *not* intended to describe every detail of BLAST
  - NCBI provides detailed documentation on BLAST
  - https://blast.ncbi.nlm.nih.gov/doc/blast-help/
- The BLAST programs described in this presentation are recent *BLAST+* programs.
- The interface of online BLAST services might be improved by anytime
  - They might look different from what was described in this presentation

Topics

- 1. Theoretical alignment algorithm
- 2. BLAST -- Basic Local Alignment Search Tools
- 3. Understanding BLAST statistics
- 4. Major variants of BLAST programs
- 5. Online BLAST services: NCBI & Ensembl
- 6. Standalone BLAST programs

Theoretical alignment algorithm

- In this section, we will go through
  - edit distances,
  - global alignments,
  - dynamic programming, and
  - local alignments.

Edit distance

- The very first question
  - "a way of quantifying how *dissimilar* two strings (e.g., words) are to one another by counting the *minimum number of operations* required to transform one string into the other"
  - In bioinformatics, *operations* are usually:
    - Insertion
    - Deletion
    - Substitution

Source: Wikipedia: Edit distance

Edit distance

- Edit distance, an example
  - **k**itten → **s**itten, one substitution
  - sitt**e**n → sitt**i**n, another substitution
  - sittin → sittin**g**, one insertion
- From “kitten” to “sitting”, we need *three* operations.
  - The distance between the two words is 3.

Source: Wikipedia: Edit distance

Global alignment

- "In bioinformatics, it can be used to quantify the *similarity* of DNA sequences, which can be viewed as strings of the letters A, C, G and T."
  - Source: Wikipedia: Edit distance

```
subject   A G C A T T C - A C A G A
          | | |   | | |   | |   | |
query     A G C - T T C G A C C G A
```

deletion insertion mismatch

Alignment length: 13
edit distance: 3
=> alignment identity
= (13-3)/13 = 77%

Global alignment

- Assuming
  - each match base gives *score* +1
  - each mismatch/insertion/deletion gives *penalty* -1
- Given the two sequences, we can have an alignment of score 4.

```
A G C A T T C A C A G A
| | |   |     | |   | |
A G C T T C G A C C G A
```

score: 4

no InDels

Global alignment

- With the same two sequence, we can have another alignment of score 7.

```
A G C A T T C - A C A G A
| | |   | | |   | |   | |
A G C - T T C G A C C G A
```

score: 7

- Question: given two sequences, how can we be sure that an alignment is of the *best* score?

Global alignment

- The dynamic programming algorithm

Ø for *null* strings

	Ø	A	G	C	A	T	T	C	A	C	A	G	A
Ø	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
A	-1	**1**	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
G	-2	0	**2**	1	0	-1	-2	-3	-4	-5	-6	-7	-8
C	-3	-1	1	**3**	**2**	1	0	-1	-2	-3	-4	-5	-6
T	-4	-2	0	2	2	**3**	2	1	0	-1	-2	-3	-4
T	-5	-3	-1	1	1	3	**4**	3	2	1	0	-1	-2
C	-6	-4	-2	0	0	2	3	**5**	4	3	2	1	0
G	-7	-5	-3	-1	-1	1	2	**4**	4	3	2	3	2
A	-8	-6	-4	-2	0	0	1	3	**5**	4	4	3	4
C	-9	-7	-5	-3	-1	-1	0	2	4	**6**	5	4	3
C	-10	-8	-6	-4	-2	-2	-1	1	3	5	**5**	4	3
G	-11	-9	-7	-5	-3	-3	-2	0	2	4	4	**6**	5
A	-12	-10	-8	-6	-4	-4	-3	-1	1	3	5	5	**7**

Dynamic programming

- The key is the incremental computation based on *previous* results on *every cell of the matrix*

	Ø	A
Ø	0	-1

"A" to Ø, a deletion score: -1

	Ø
Ø	0
A	-1

Ø to "A", an insertion score: -1

	Ø	A
Ø	0	-1
A	-1	**1**

"A" to "A", three possibilities
1. delete A and insert A: score -2 (green)
2. insert A and delete A: score -2 (blue)
3. match of A: score 1 (red, the best)

Dynamic programming

- The incremental computation for the entire matrix

Ø for *null* strings

	Ø	A	G	C	A	T	T	C	A	C	A	G	A
Ø	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
A	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
G	-2	0	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
C	-3	-1	1	3	2	1	0	-1	-2	-3	-4	-5	-6
T	-4	-2	0	2	2	3	2	1	0	-1	-2	-3	-4
T	-5	-3	-1	1	1	3	4	3	2	1	0	-1	-2
C	-6	-4	-2	0	0	2	3	5	4	3	2	1	0
G	-7	-5	-3	-1	-1	1	2	4	4	3	2	3	2
A	-8	-6	-4	-2	0	0	1	3	5	4	4	3	4
[illegible]	[illegible]	[illegible]	[illegible]	-3	-1	-1	0	2	4	6	5	4	3
[illegible]	[illegible]	[illegible]	[illegible]	-4	-2	-2	-1	1	3	5	5	4	3
[illegible]	[illegible]	[illegible]	[illegible]	-5	-3	-3	-2	0	2	4	4	6	5
A	-12	-10	-8	-6	-4	-4	-3	-1	1	3	5	5	7

The computational time would be *proportional* to the *size of the matrix*

Dynamic programming

- Trace *back* and get the global alignment

Ø for *null* strings

	Ø	A	G	C	A	T	T	C	A	C	A	G	A
Ø	-	D	D	D	D	D	D	D	D	D	D	D	D
A	I	M	D	D	M	D	D	D	M	D	M	D	M
G	I	I	M	D	D	D	D	D	D	D	D	M	D
C	I	I	I	M	D	D	D	M	D	M	D	D	D
T	I	I	I	I	S	M	M	D	D	D	D	D	D
T	I	I	I	I	S	M	M	D	D	D	D	D	D
C	I	I	I	M	S	I	I	M	D	M	D	D	D
G	I	I	M	I	S	I	I	I	S	S	S	M	D
A	I	M	I	I	M	I	I	I	M	D	M	D	M
C	I	I	I	M	I	S	I	M	I	M	D	D	D
C	I	I	I	M	I	S	I	M	I	M	S	S	S
G	I	I	M	I	I	S	I	I	I	I	S	M	D
A	I	M	I	I	M	S	I	I	M	I	M	I	M

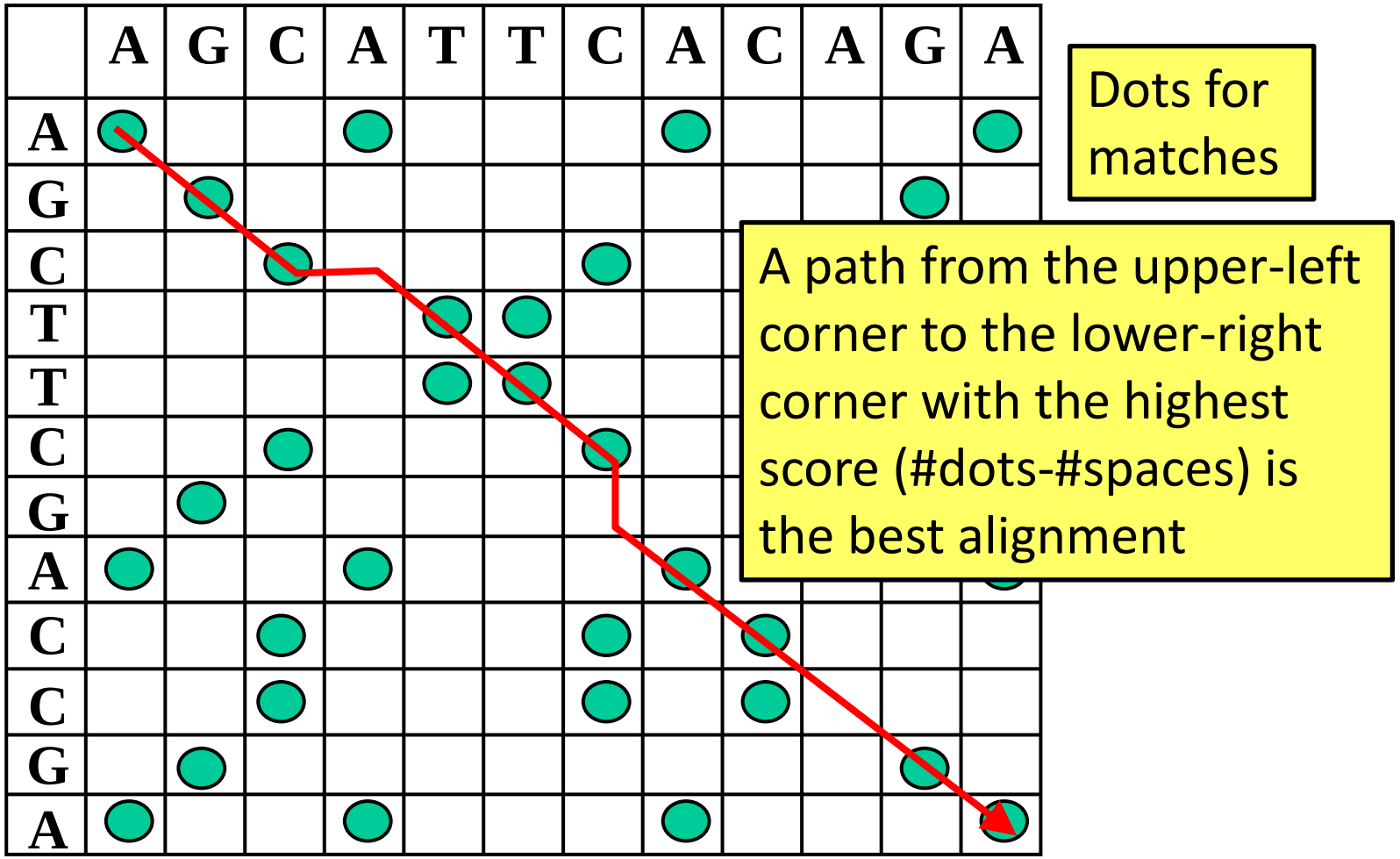
M: match
S: substitution
I: insertion
D: deletion

Dynamic programming

- You may check sheet DynamicProgramming of the supplement Excel file LocalAlignmentExample_20250513.xlsx for Excel formulas of dynamic programming for string alignments

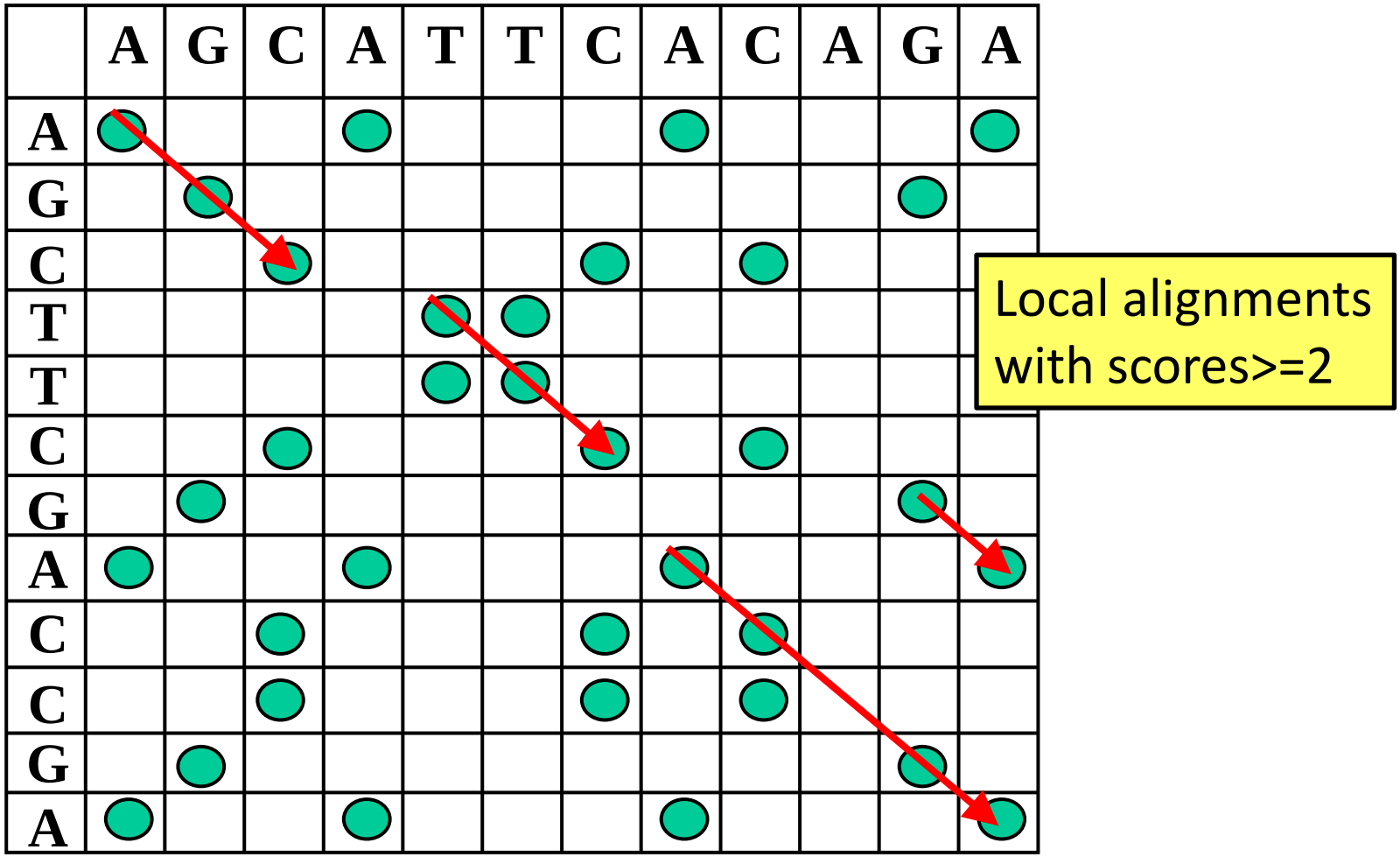
Global alignment

- Dot plots, another way of viewing the matrix



Local alignment

- Dotplots – local alignments, diagonals of dots



Local alignment

- Physical meaning of
  - Global alignments
    - Determine if two sequences are *entirely* similar to each other.
  - Local alignments
    - Find *subsequences* in one sequence that are very similar to *subsequences* in the other sequence.
    - To find *any* similarity from one sequence (query) in some other sequences (database)

Local alignment

- Exact local alignment search algorithms
  - Step 1: build up a table of size $m \times n$ (m: length of query, n: length of database) like a dot plot
  - Step 2: search diagonals of dots (local alignments)
- This kind of approaches would cost at least $m \times n$ units of time.
  - Image that that the database might be something like nr/nt, RefSeq, UniProt …
    - should be *time-consuming*

Short summaries

- In computer science, edit distances are used for measuring the distance between two strings.
- In bioinformatics, *scores of matches + penalties of insertion/deletion/substation* are for measurement of the similarity between two sequences.

Short summaries

- Dynamic programming helps to find
  - global alignments and
  - local alignments
  - for entire sequences and subsequences, respectively.
- Time cost of dynamic programming is proportional to the size of query *times* the size of target database.
  - could be *time-consuming* for large databases.

BLAST

- Basic Local Alignment Search Tool
  - a *heuristic* algorithm
    - BLAST assumes local alignments to be found containing exact matches no less than *W*
    - Consider an alignment as a series of coin tossing with outcomes Head (match) or Tail (mismatch/InDel)
    - High similarity means a long run of Heads (matches)

```
A G C A T T C - A C A G A
| | |   | | |   | |   | |
A G C - T T C G A C C G A
H H H T H H H T H H T H H
```

alignment identity = 77%
longest run of matches = 3

BLAST

- The algorithm largely reduces the search time by
  - building a look-up table that stores all positions of *W*-mers of the query sequence
    - takes *m* units of time
  - looking for these *W*-mers in the database
    - takes *n* units of time
  - forming local alignments based on the *W*-mer seeds

BLAST

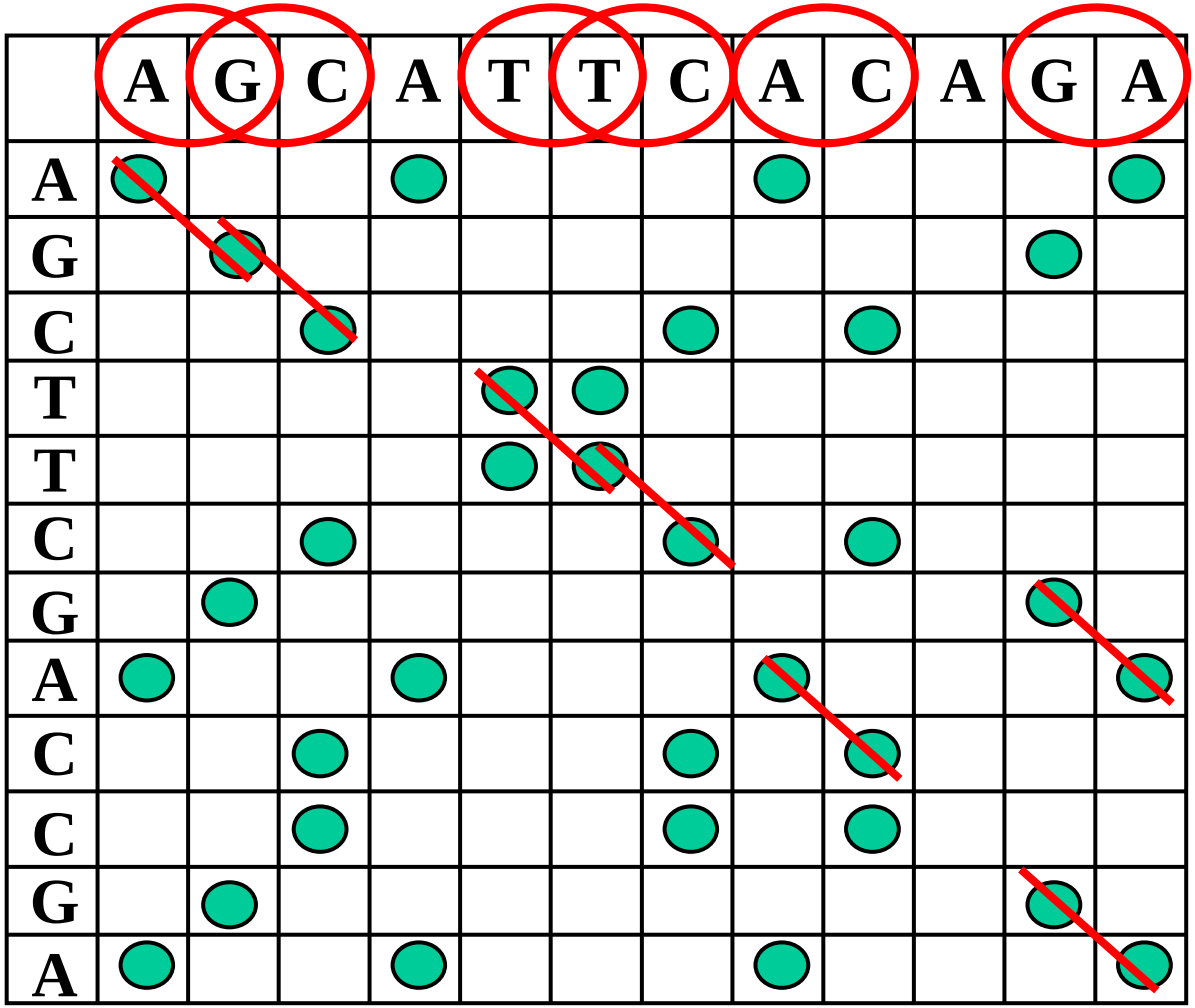
- Build a look-up table of all 2-mers
  - Query sequence: AGCTTCGACCGA
  - *W*=2

AG	1
GC	2
CT	3
TT	4
TC	5
CG	6,10
GA	7,11
AC	8
CC	9

BLAST

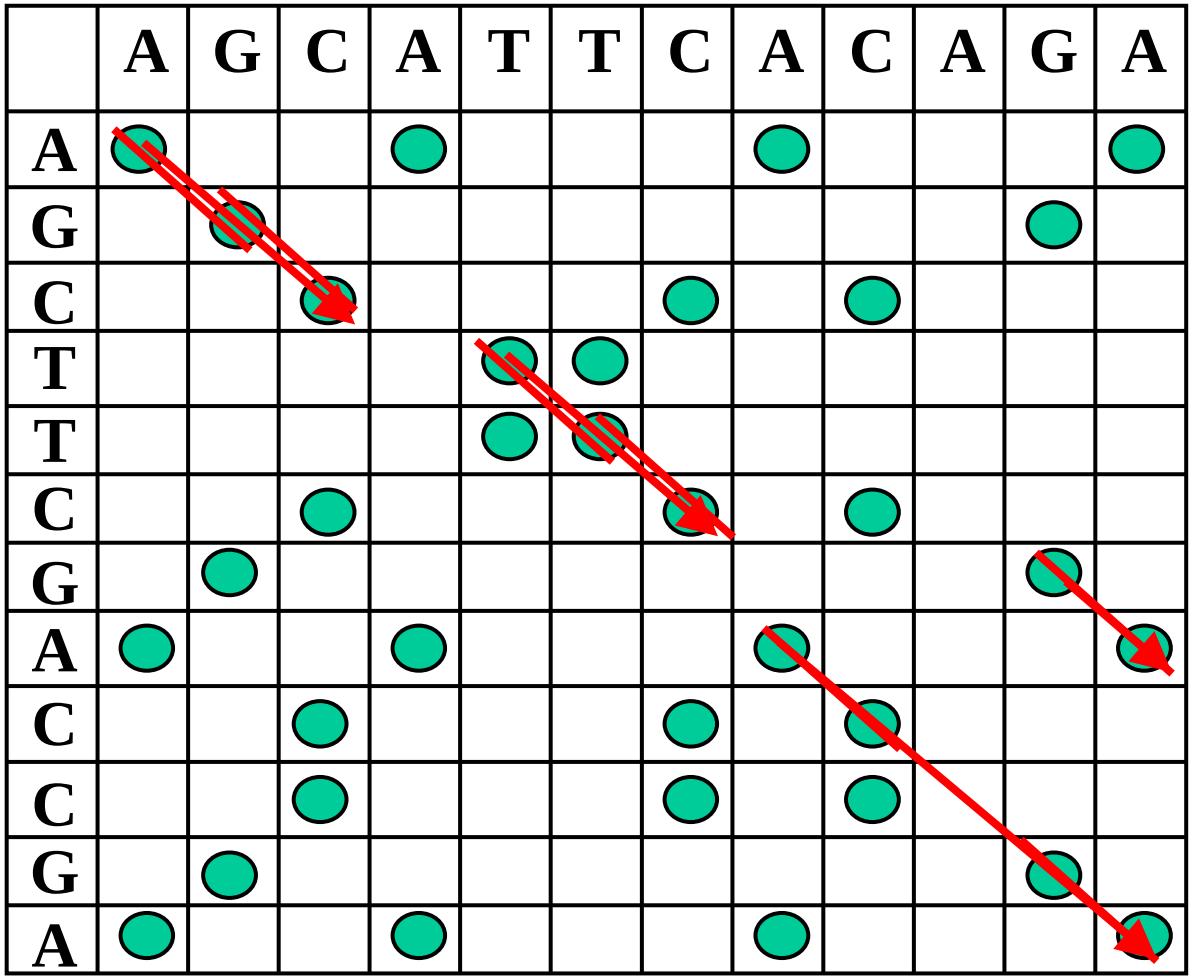
- look for these *W*-mers in the database

AG	1
GC	2
CT	3
TT	4
TC	5
CG	6,10
GA	7,11
AC	8
CC	9



BLAST

- Form local alignments based on the *W*-mer-hits



Short summaries

- Time cost of theoretical dynamic programming is proportional to the size of query *times* the size of target database.
- The BLAST *heuristic* algorithm *assumes* consecutive *W* matches in the result alignments
  - The high similarity => *usually* the longer consecutive matches

Short summaries

- The time cost of BLAST is proportional to the *sum* of
  - the size of query (for building the look-up table),
  - the size of database (scanning against the look-up table), and
  - the volume of total alignments (which is usually very small, compared to the size of database).
- Much *smaller* than that of dynamic programming.

BLAST statistics

- In addition to the fast heuristic, BLAST computes an E-value for each alignment
  - based on the similarity score

A minor constant

Scaling factor

E-value

$$E = kmne^{-\lambda s}$$

Raw score

Length of query

Length of database

$m \times n$: the search space

BLAST statistics

- The E-value means the expected number of local alignments that have alignment scores greater than or equal to *S* in *this* BLAST search.
  - An E-value close to zero means that an alignment with score *S* or greater is *not likely* to appear in a random sequence model.
    - The alignment should not be “random.”
    - A thinking of statistical hypothesis testing.

BLAST statistics

- For each local alignment, there will be a summary like this

hit sequence in database

```
>swissprot:CTRB_HUMAN Chymotrypsinogen B precursor (EC 3.4.21.1).
          Length = 263

 Score =  433 bits (1222), Expect = e-121
 Identities = 220/263 (83%), Positives = 252/263 (95%), Gaps = 2/263 (0%)
```

bit score **raw score** **E value**

bit scores are *normalized* scores

BLAST statistics

- Within one BLAST search
  - It is feasible to compare alignments based on E-values or bit scores.
  - A smaller (more significant) E-value means an alignment *less* likely to be “random.”
  - A larger bit score means an alignment of two “closer” subsequences.

BLAST statistics

- From different runs of BLAST searches, you may compare alignments based on
  - bit score, the normalized score.
  - E-values are no longer feasible to be used for comparing alignments
    - Recall that $E = kmne^{-\lambda S}$
    - if the same query sequence (length m) were used to search against different databases (length $n_1 \neq n_2$, respectively)
    - alignments with the same score S would result in different E-values $kmn_1e^{-\lambda S} \neq kmn_2e^{-\lambda S}$

Short summaries

- Knowing BLAST statistics better would help you to interpret BLAST outputs better.
- A way to fix "the same score but different *E*-values in different BLAST searches" problem when running *standalone* BLAST programs
  - specify a fixed *effective search space* (i.e. $m \times n$ in the *E*-value formula)
  - option "-searchsp" for BLAST+ programs

Major variants of BLAST programs

- BLAST**N**
  - Searching **n**ucleotide databases using a **n**ucleotide query.
  - The underlying algorithm should be closed to what we described in the BLAST algorithm section.
    - Build a *W*-mer look-up table of the query
    - Scan the database against the look-up table, a hit was identified if an exact match
    - Form alignments based on *W*-mer hits in the database

Major variants of BLAST programs

- BLAST**P**
  - Searching **p**rotein databases using a **p**rotein query.
  - The underlying algorithm should be *similar* with what we described in the BLAST algorithm section
    - Build a *W*-mer look-up table of the query
    - Scan the database against the look-up table, a seed was identified if the database *W*-mer is close enough to the query *W*-mer, given the *AA-to-AA scoring matrix*.
    - Form alignments based on *W*-mer hits in the database. Alignment scores were computed according to the *AA-to-AA scoring matrix*

Major variants of BLAST programs

- BLAST**X**
  - Searching protein databases using a nucleotide query by translating 1 query into 6 protein queries using the six reading frames.
  - Actual sequence search was done like using BLASTP.
  - Considering this as running 6 times of BLASTP.

Major variants of BLAST programs

- **T**BLAST**N**
  - Searching nucleotide databases using a protein query by translating the database using the six reading frames.
  - Actual sequence search was done like using BLASTP.
  - Considering this as running 6 times of BLASTP.

Major variants of BLAST programs

- **T**BLAST**X**
  - Searching nucleotide databases using a nucleotide query by translating *both* query and the database using the six reading frames.
  - Actual sequence search was done like using BLASTP.
  - Considering this as running 36(=6x6) times of BLASTP.

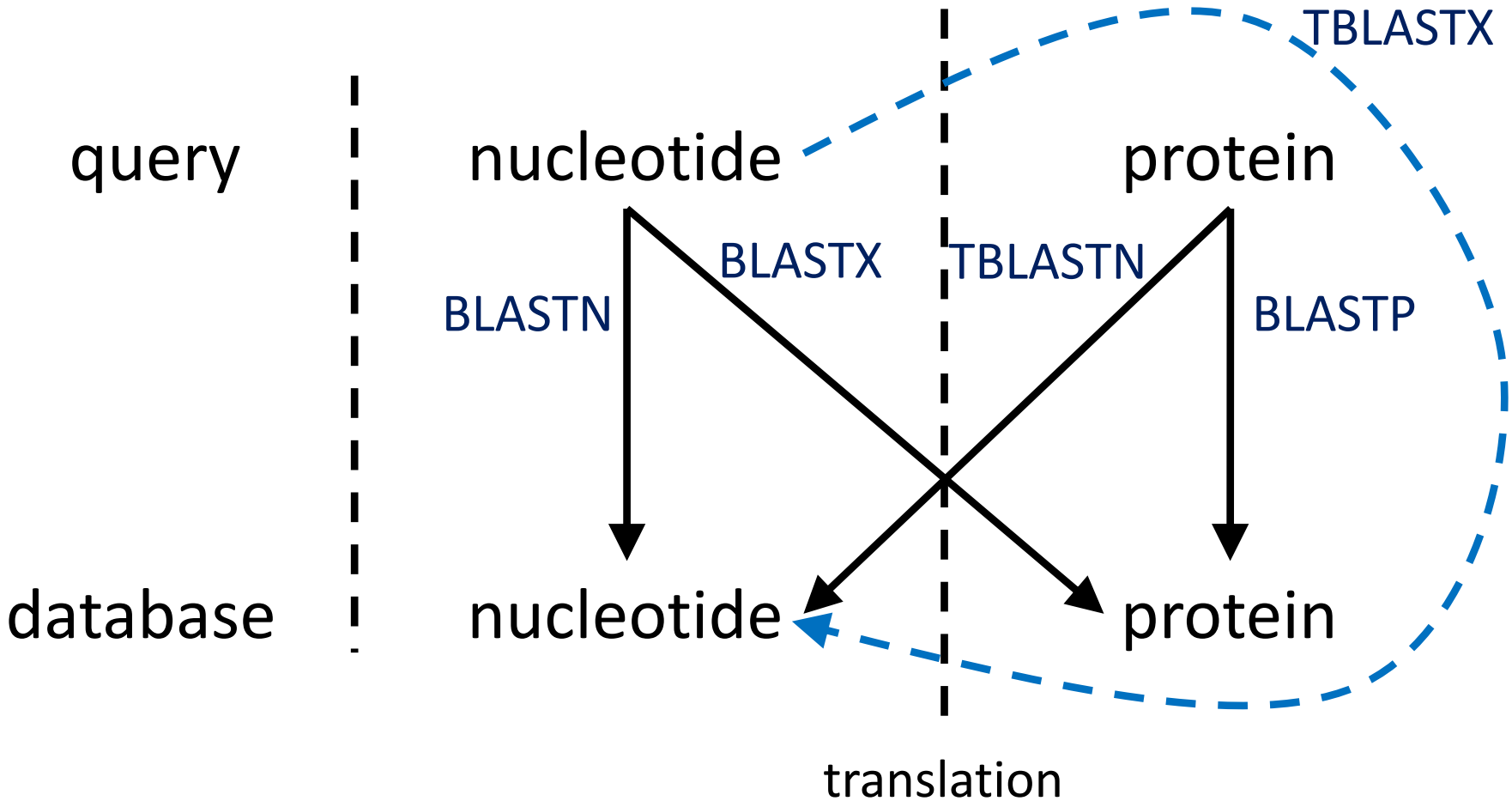
Major variants of BLAST programs

- If the actual alignment to be done is for protein sequences (blastp, blastx, tblastn, and tblastx),
  - there will be a matrix parameter for scoring.

Matrix	Best use	Similarity (%)
BLOSUM90	Short alignments that are highly similar	70-90
BLOSUM80	Detecting members of a protein family	50-60
BLOSUM62	Most effective in finding all potential similarities	30-40
BLOSUM30	Longer alignments of more divergent sequences	<30

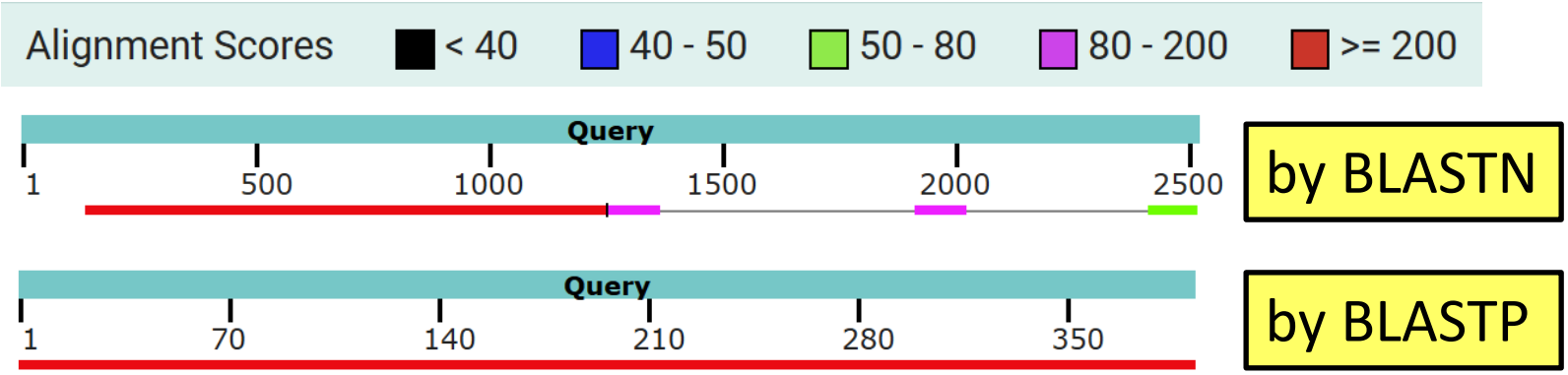
Short summaries

- The five major BLAST variants



A short note

- From above, we can find that BLASTN is the only BLAST variant that compares nucleotide sequences in the underlying level.
  - Comparing protein sequences would be *more sensitive* than comparing nucleotide sequences.
  - Example: comparing human-mouse TP53



Online BLAST services

- In this section, we will demonstrate usages of online BLAST services provided by NCBI and Ensembl
  - NCBI: https://blast.ncbi.nlm.nih.gov/Blast.cgi
  - Ensembl (plants): https://plants.ensembl.org/Multi/Tools/Blast
  - These service sites are periodically updating their functionalities
    - descriptions here might be a little different with the actual pages *later*

Various BLAST programs

NCBI BLAST services

- https://blast.ncbi.nlm.nih.gov/Blast.cgi
  - or google "NCBI BLAST"

Help docs

BLAST ®

Home Recent Results Saved Strategies Help

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. Learn more

NEWS

BLAST+ 2.13.0 is here!

Starting with this release, we are including the blastn_vdb and tblastn_vdb executables in the BLAST+ distribution.

Thu, 17 Mar 2022 12:00:00 EST

More BLAST news...

Various BLAST programs

Web BLAST

Nucleotide BLAST
nucleotide ▶ nucleotide

blastx
translated nucleotide ▶ protein

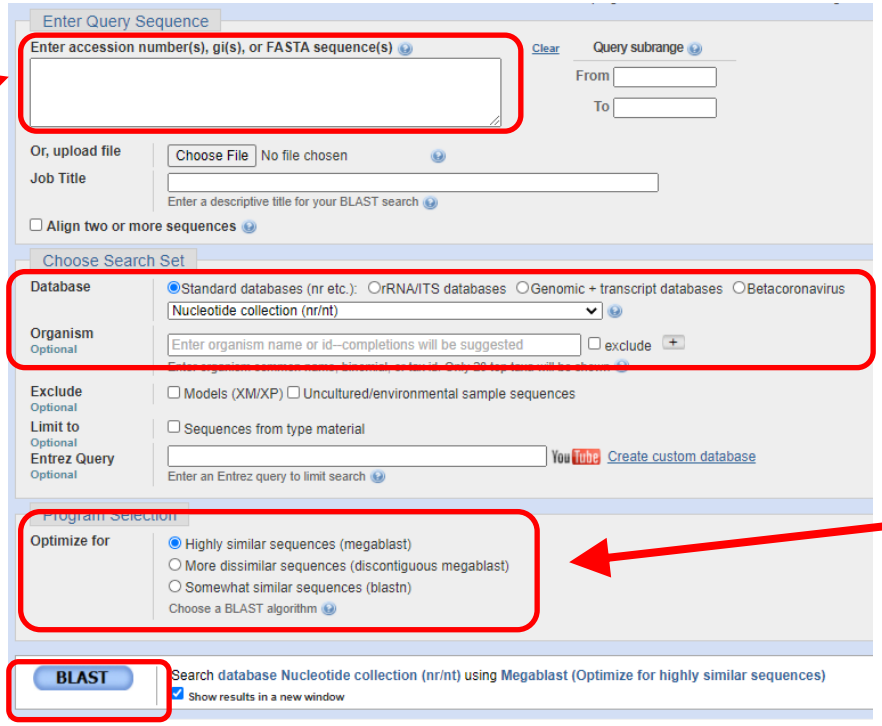
tblastn
protein ▶ translated nucleotide

Protein BLAST
protein ▶ protein

NCBI BLAST services

- Entering any BLAST program in last slide would lead you to pages of similar organization
  - Take Nucleotide BLAST as an example

Enter query sequence(s)



Pick target database, and optionally set organism constraint

Finer program selection

Run!

NCBI BLAST services

- query sequence(s)
  - Can be in multi-FASTA formats or bare sequence

the FASTA format, can be of multiple query sequenes

```
>seqA
GACCATGGCGGAGGAATTTGGAAGCATAGATTTACTCGGAGATGAAGATT
TCTTCTTCGATTTCGATCCTTCAATCGTAATTGATTCTCTTCCGGCGGAG
>seqB
GATTTTCTTCAGTCTTCACCGGATTCATGGATCGGAGAAATCGAGAATCA
ATTGATGAACGATGAGAATCATCAAGAGGAGAGTTTTGTGGAATTGGATC
AGCAATCGGTTTCAGATTTCATAGCGGATCTACTCGTTGATTATCCAACT
```

Single query sequence

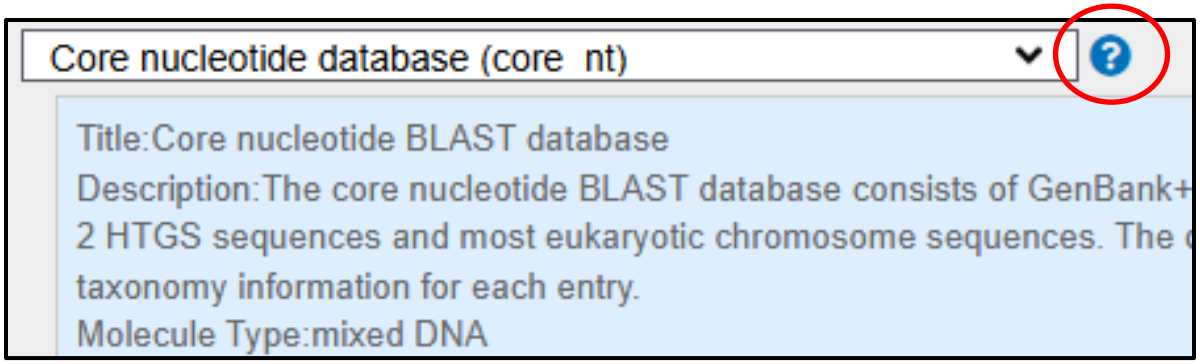
```
  1 GACCATGGCG GAGGAATTTG GAAGCATAGA TTTACTCGGA GATGAAGATT
 51 TCTTCTTCGA TTTCGATCCT TCAATCGTAA TTGATTCTCT TCCGGCGGAG
101 GATTTTCTTC AGTCTTCACC GGATTCATGG ATCGGAGAAA TCGAGAATCA
151 ATTGATGAAC GATGAGAATC ATCAAGAGGA GAGTTTTGTG GAATTGGATC
201 AGCAATCGGT TTCAGATTTC ATAGCGGATC TACTCGTTGA TTATCCAACT
```

```
GACCATGGCGGAGGAATTTGGAAGCATAGATTTACTCGGAGATGAAGATT
TCTTCTTCGATTTCGATCCTTCAATCGTAATTGATTCTCTTCCGGCGGAG
GATTTTCTTCAGTCTTCACCGGATTCATGGATCGGAGAAATCGAGAATCA
ATTGATGAACGATGAGAATCATCAAGAGGAGAGTTTTGTGGAATTGGATC
AGCAATCGGTTTCAGATTTCATAGCGGATCTACTCGTTGATTATCCAACT
```

All these three formats are feasible

NCBI BLAST services

- Target database
  - Usually we pick (core) nr/nt or Refseq
    - (core) nt: GenBank+EMBL+DDBJ+PDB+RefSeq sequences, but excludes EST, STS, GSS, WGS, TSA. *Identical* sequences have been merged into one entry.
    - refseq_rna: RNA parts of RefSeq. RefSeq is a comprehensive, integrated, non-redundant, well-annotated set of sequences, including genomic DNA, transcripts, and proteins. (ref: NCBI RefSeq)



make a good use of those question icons!

A short note

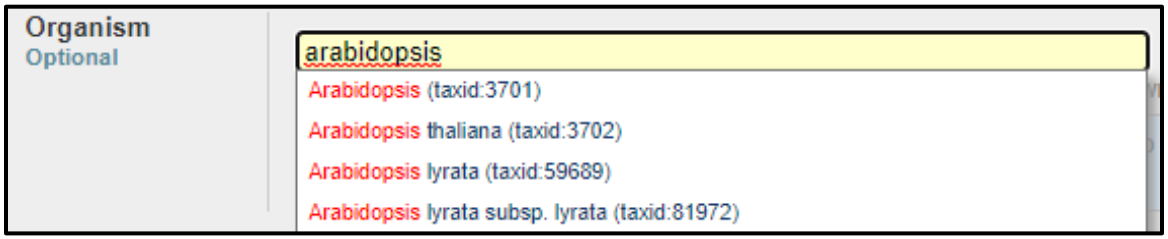
- Why pick (core) nr/nt?
  - nr for the non-redundant collection of proteins and nt for the non-redundant collection of nucleotides.
    - They are representing *gene-level universal collections* of sequences in NCBI.
  - Searching against nr/nt means we usually got a hit if there is a similar sequence in NCBI

A short note

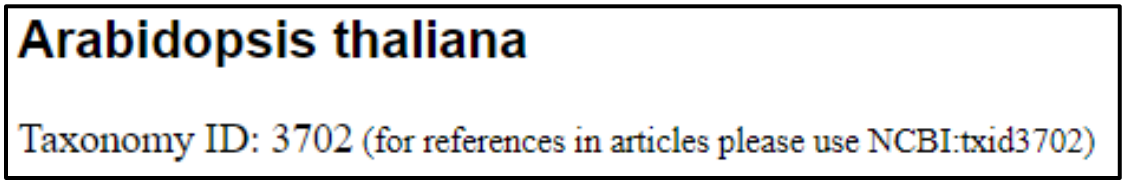
- Why pick Refseq rna/protein?
  - Refseq rna/protein datasets were made by collecting sequences of *curated* transcriptomes and proteomes of complete genomes
  - Searching against Refseq rna/protein under some species constraint *usually* means that no gene would be missed for those species.
- Checking database descriptions help you to make decisions!

NCBI BLAST services

- Set organism constraint
  - Entering name of your target organism would bring out a pull-down menu, simply pick the desired organism.

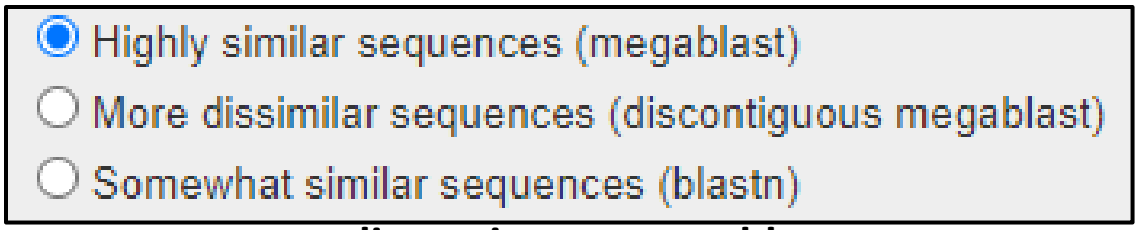


  - A precise way is to find out taxid or the target organism from the NCBI Taxonomy database
    - google "NCBI Taxonomy"

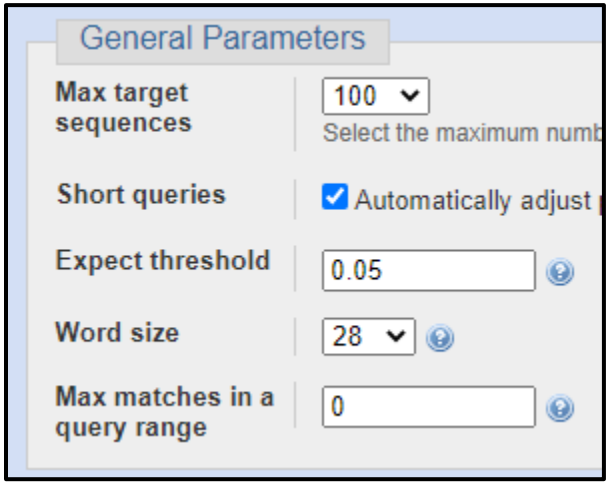


NCBI BLAST services

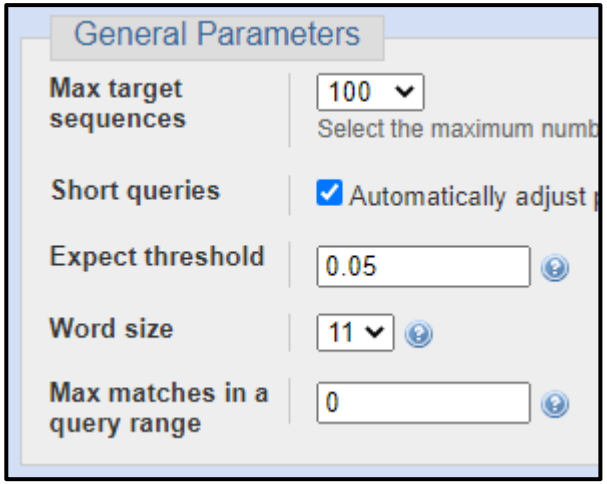
- Finer program selection
  - For Nucleotide BLAST, there actually a few number of variations of programs doing the similar tasks
  - The key difference between them are parameters



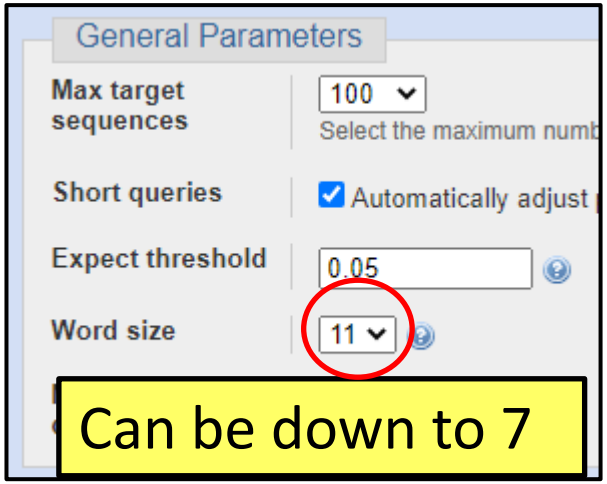
megablast



discontiguous megablast

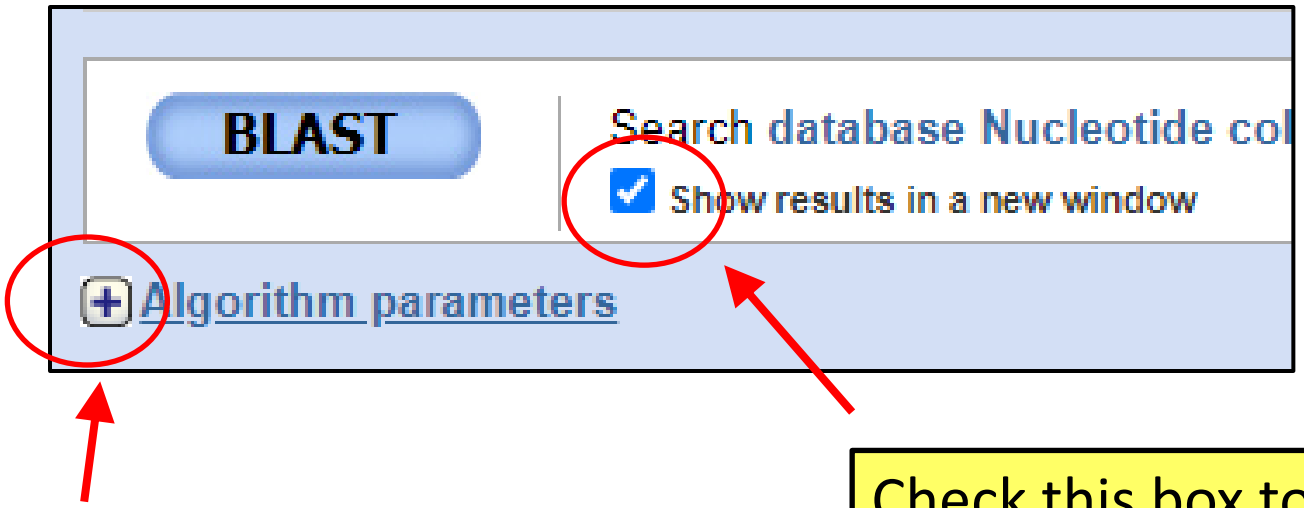


blastn



NCBI BLAST services

- Before you run BLAST



Expand this to check or adjust parameters

Check this box to keep the input page available and create a new page for BLAST progress and results

NCBI BLAST services

- A BLASTN example
  - In this example, we would like to search Arabidopsis thaliana bZIP60 against NCBI RefSeq database
  - bZIP60 transcript sequence
    - https://www.arabidopsis.org/sequence?key=1002431237

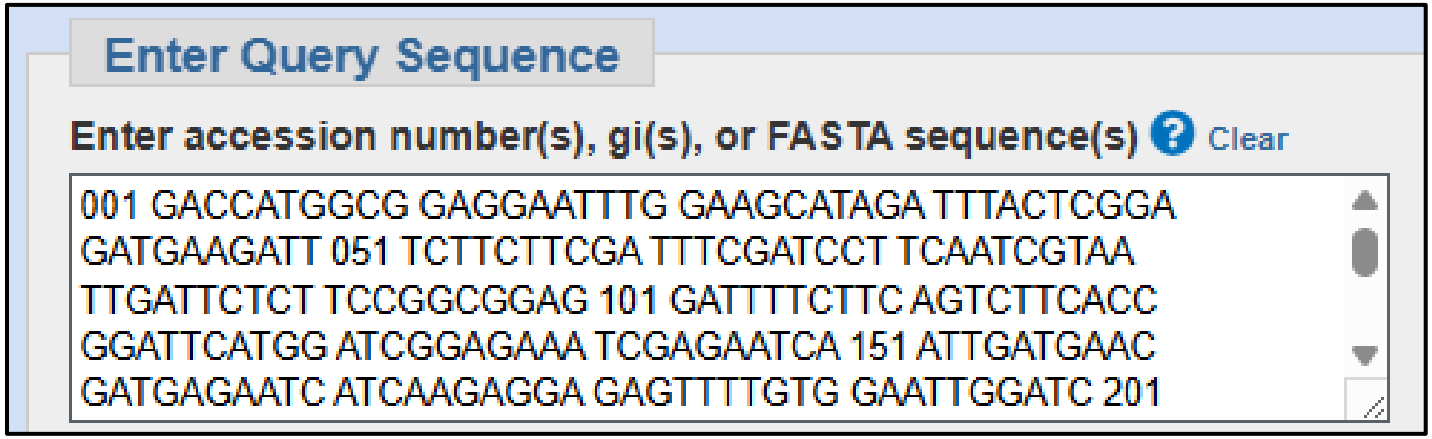
NCBI BLAST services

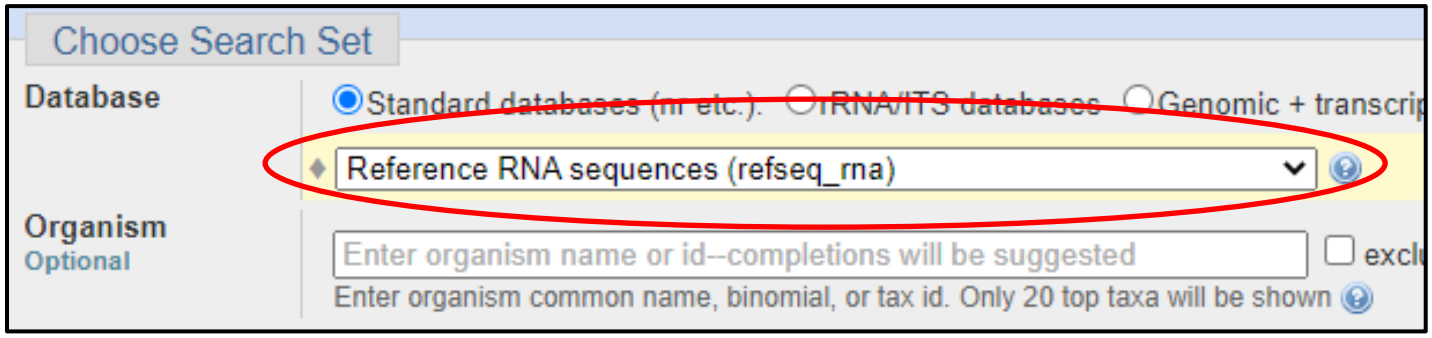
- The bZIP60 transcript sequence

```
001 GACCATGGCG GAGGAATTTG GAAGCATAGA TTTACTCGGA GATGAAGATT
051 TCTTCTTCGA TTTCGATCCT TCAATCGTAA TTGATTCTCT TCCGGCGGAG
101 GATTTTCTTC AGTCTTCACC GGATTCATGG ATCGGAGAAA TCGAGAATCA
151 ATTGATGAAC GATGAGAATC ATCAAGAGGA GAGTTTTGTG GAATTGGATC
201 AGCAATCGGT TTCAGATTTC ATAGCGGATC TACTCGTTGA TTATCCAACT
251 AGCGATTCTG GCTCCGTTGA TTTGGCGGCT GATAAAGTTC TAACCGTCGA
301 TTCTCCCGCC GCCGCTGATG ATTCCGGGAA GGAGAATTCG GATTTGGTTG
351 TTGAGAAGAA GTCTAATGAT TCTGGTAGCG AGATTCATGA TGATGATGAC
401 GAAGAAGGAG ACGATGATGC TGTGGCTAAA AAACGAAGAA GGAGAGTAAG
451 AAATAGAGAT GCGGCGGTTA GATCGAGAGA GAGGAAGAAG GAATATGTAC
501 AAGATTTAGA GAAGAAGAGT AAGTATCTCG AAAGAGAATG CTTGAGACTA
551 GGACGTATGC TTGAGTGCTT CGTTGCTGAA AACCAGTCTC TACGTTACTG
601 TTTGCAAAAG GGTAATGGCA ATAATACTAC CATGATGTCG AAGCAGGAGT
651 CTGCTGTGCT CTTGTTGGAA TCCCTGCTGT TGGGTTCCCT GCTTTGGCTT
701 CTGGGAGTAA ACTTCATTTG CCTATTCCCT TATATGTCCC ACACAAAGTG
751 TTGCCTCCTA CGTCCAGAAC CAGAAAAGCT GGTTCTAAAC GGGCTCGGGA
801 GTAGTAGCAA ACCGTCTTAT ACCGGCGTTA GTCGGAGATG TAAGGGTTCG
851 AGGCCTAGGA TGAAATACCA AATCTTAACC CTTGCGGCGT GACAACGCCT
901 TTTTTAACTG CTTCTTTTGC GCATTTTGAG TTGTAGATGA GTGTCTTTTA
951 GTTTTCTCTC TCTTGTTTTG TATTTCGCTG TTGAAAGTTT TCTGTCTAAT
1001 ATCGATAAGT TAACAGTGAA TGTGGGTCTT ATGGTTATGG ATGATATCTA
1051 TCTAATAATG CTTCTGCCTT TAAAATGTTG ATTTTGAGGC ATAACTTCAG
1101 GTAATATCAC TTCTAATTAC TAGATAACAA TTCATTAGGT TGATTAACAT
1151 TGATAAAGCT TTTCCTCATG CTAGTTTTTA CATGTTTGCT TCATTTGACA
1201 TTATCACAGT TTTTTTTTTT TTTTTTTTTG T
```

NCBI BLAST services

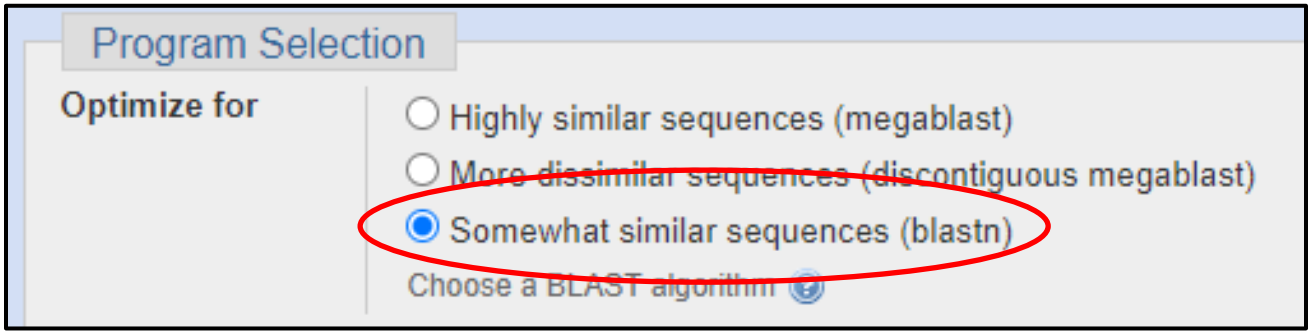
- In Nucleotide BLAST page

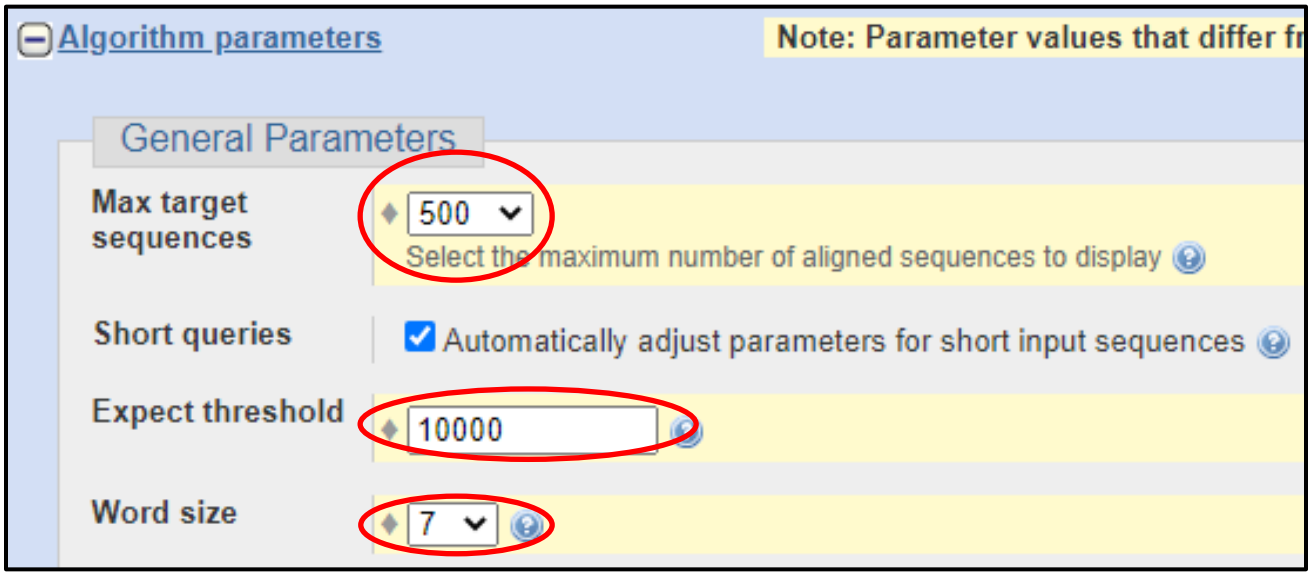




NCBI BLAST services

- In Nucleotide BLAST page





NCBI BLAST services

- In the result page

BLAST ® » blastn suite » results for RID-22RB7DSE013

< Edit Search | Save Search | Search Summary

Job Title	Nucleotide Sequence
RID	22RB7DSE013 *Search expires on 05-13 10:31 am* Download All
Program	BLASTN Citation
Database	refseq_rna See details
Query ID	lcl|Query_4858153
Description	None
Molecule type	dna
Query Length	1231
ports	Distance tree of results

Search info

NCBI will keep the search result for a few days. Retrieve the result according to the RID. Or simply download the result.

NCBI BLAST services

- In the result page
  - The Descriptions tab gives descriptions of hit sequences.

pick columns to show

Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments Download Select columns Show 500

select all *500 sequences selected* GenBank Graphics Distance tree of results MSA Viewer

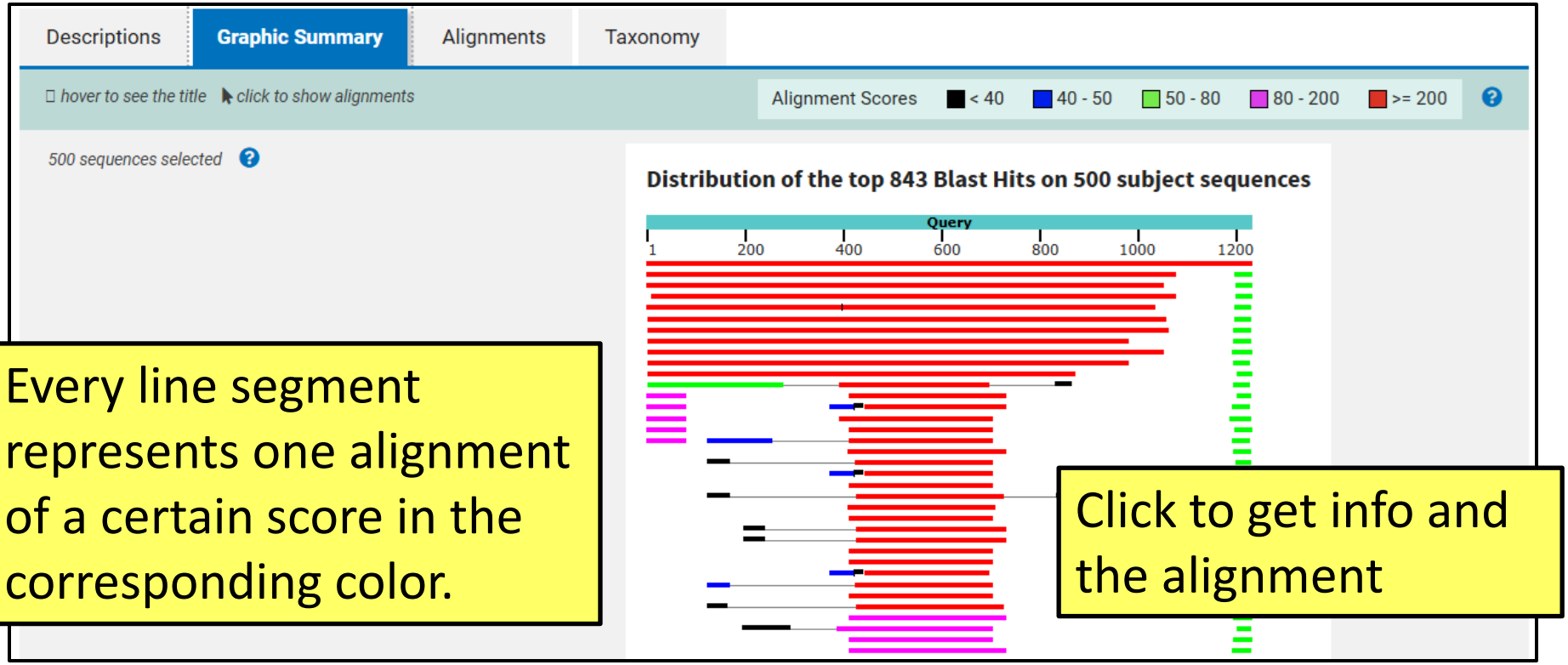
Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Arabidopsis thaliana basic region/leucine zipper motif 60 (BZIP60), mRNA	Arabidopsis thal...	2221	2221	100%	0.0	100.00%	1231	NM_103458.3
PREDICTED: Arabidopsis lyrata subsp. lyrata bZIP transcription factor 60 (LOC9327309), mRNA	Arabidopsis lyra...	1499	1499	87%	0.0	90.04%	1195	XM_021013469.1
PREDICTED: Capsella rubella bZIP transcription factor 60 (LOC17897547), mRNA	Capsella rubella	1077	1077	85%	0.0	82.23%	1118	XM_006305391.2
PREDICTED: Camelina sativa bZIP transcription factor 60 (LOC104779156), mRNA	Camelina sativa	1042	1042	87%	0.0	80.35%	1315	XM_010503541.1
PREDICTED: Camelina sativa bZIP transcription factor 60 (LOC104757902), mRNA	Camelina sativa	836	1181	84%	0.0	85.83%	1256	XM_010480685.2
PREDICTED: Eutrema salsugineum bZIP transcription factor 60 (LOC18012617), mRNA	Eutrema salsugi...	764	764					395937.2

click to get alignments

click to get hit sequence info

NCBI BLAST services

- In the result page
  - The Graphic Summary tab gives alignment locations in the query side.



NCBI BLAST services

- In the result page
  - The Taxonomy tab gives taxonomy distribution of hits.

Descriptions | Graphic Summary | Alignments | **Taxonomy**

Reports | **Lineage** | Organism | Taxonomy

500 sequences selected

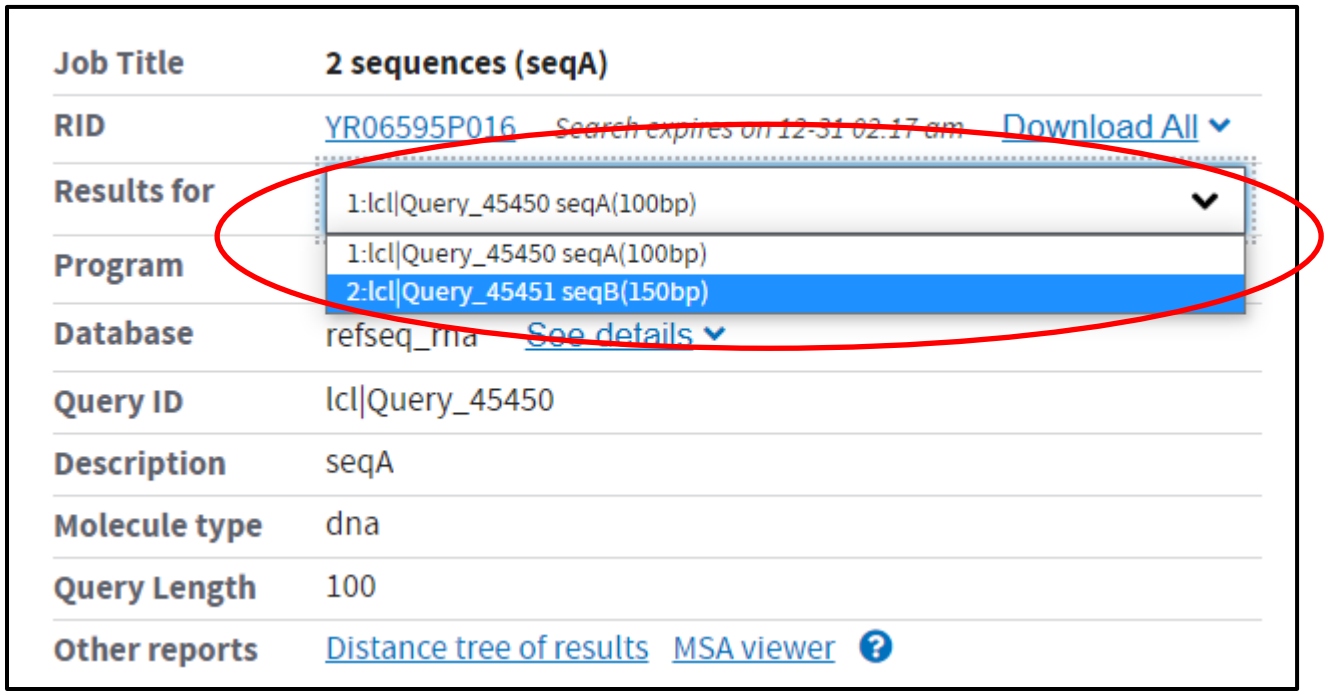
Organism	Blast Name	Score	Number of Hits	Description
Eukaryota	eukaryotes		500	
. Magnoliopsida	flowering plants		272	
. . Mesangiospermae	flowering plants		270	
. . . Pentapetalae	eudicots		211	
. . . . rosids	eudicots		145	
. . . . . malvids	eudicots		60	
. . . . . Brassicales	eudicots		21	
. . . . . Brassicaceae	eudicots		17	
. . . . . Camelineae	eudicots		11	
. . . . . Arabidopsis	eudicots		2	
. . . . . Arabidopsis thaliana	eudicots	2221	1	Arabidopsis thaliana hits
. . . . . Arabidopsis lyrata subsp. lyrata	eudicots	1499	1	Arabidopsis lyrata subsp
. . . . . Capsella rubella	eudicots	1077	1	Capsella rubella hits
. . . . . Camelina sativa	eudicots	1042	8	Camelina sativa hits

click to get hit sequence list

click to get alignment list

NCBI BLAST services

- If there are two or more query sequences,
  - there will be a pull-down menu for selecting results of different queries.



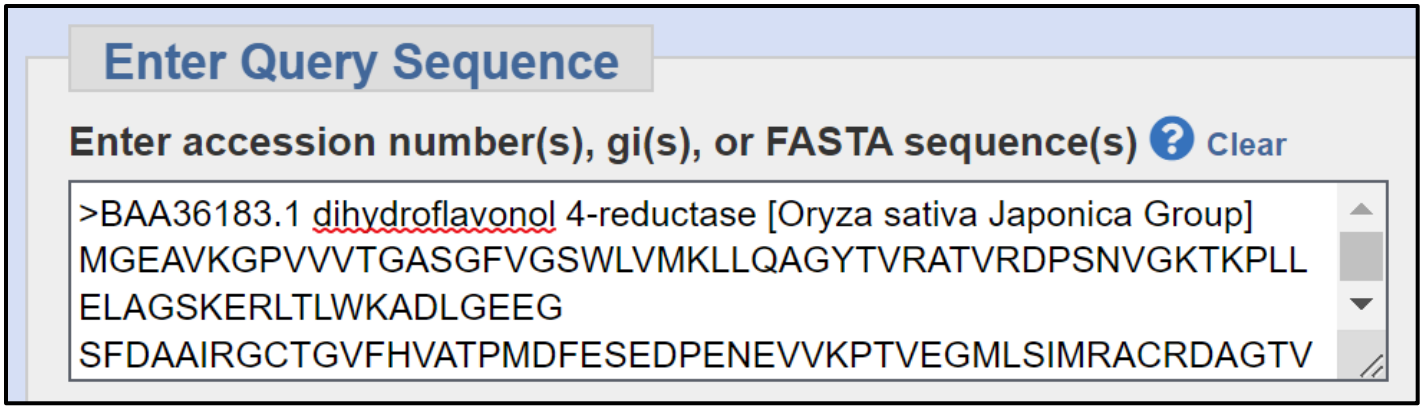
NCBI BLAST services

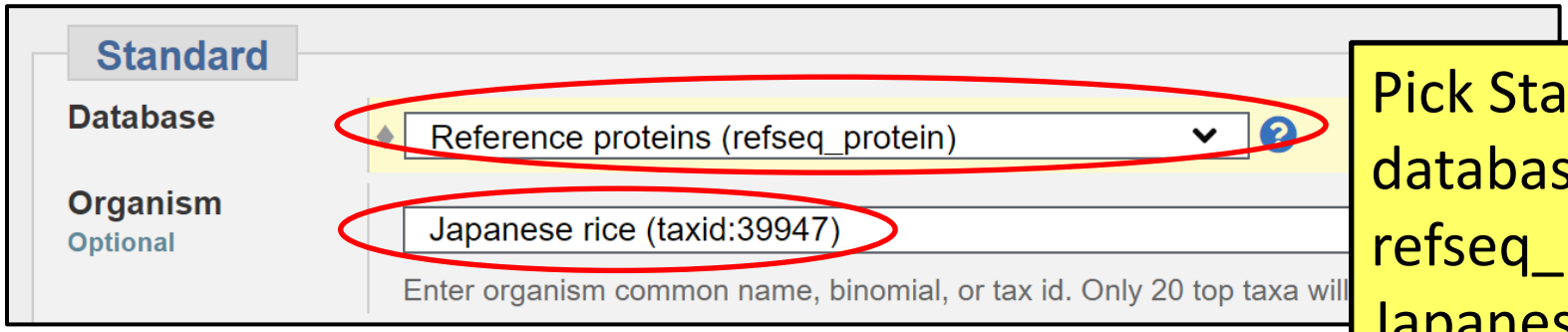
- A BLASTP example
  - In this example, we would like to search a rice *may-be-true* protein sequence BAA36183.1 (an NCBI genbank accession)

```
>BAA36183.1 dihydroflavonol 4-reductase [Oryza sativa Japonica Group]
MGEAVKGPVVVTGASGFVGSWLVMKLLQAGYTVRATVRDPSNVGKTKPLLELAGSKERLTLWKADLGEEG
SFDAAIRGCTGVFHVATPMDFESEDPENEVVKPTVEGMLSIMRACRDAGTVKRIVFTSSAGTVNIEERQR
PSYDHDDWSDIDFCRRVKMTGWMYFVSKSLAEKAAMEYAREHGLDLISVIPTLVVGPFISNGMPPSHVTA
LALLTGNEAHYSILKQVQFVHLDDLCDAEIFLFESPEARGRYVCSSHDATIHGLATMLADMFPEYDVPRS
FPGIDADHLQPVHFSSWKLLAHGFRFRYTLEDMFEAAVRTCREKGLLPPLPPPPTTAVAGGDGSAGVAGE
KEPILGRGTGTAVGAETEALVK
```

NCBI BLAST services

- In BLASTP page

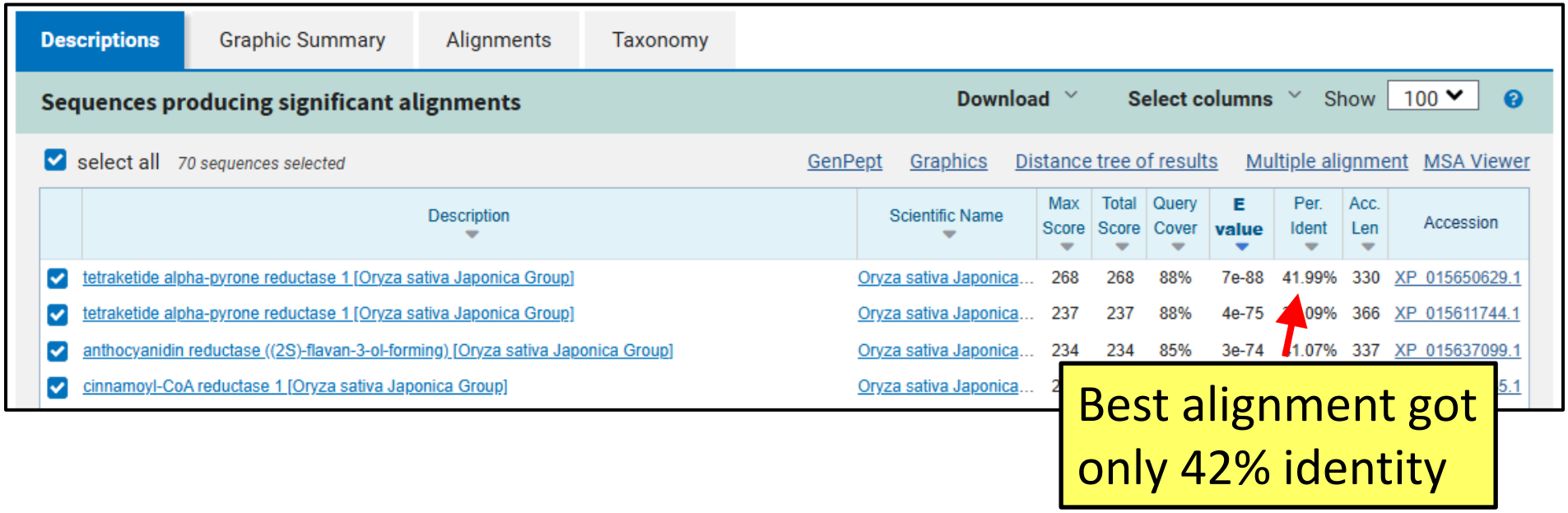




Pick Standard database, refseq_prot, and Japanese rice

NCBI BLAST services

- In this example, we are not getting an 100% identity alignment by querying this rice protein against rice proteins in the NCBI refseq_prot database.



NCBI BLAST services

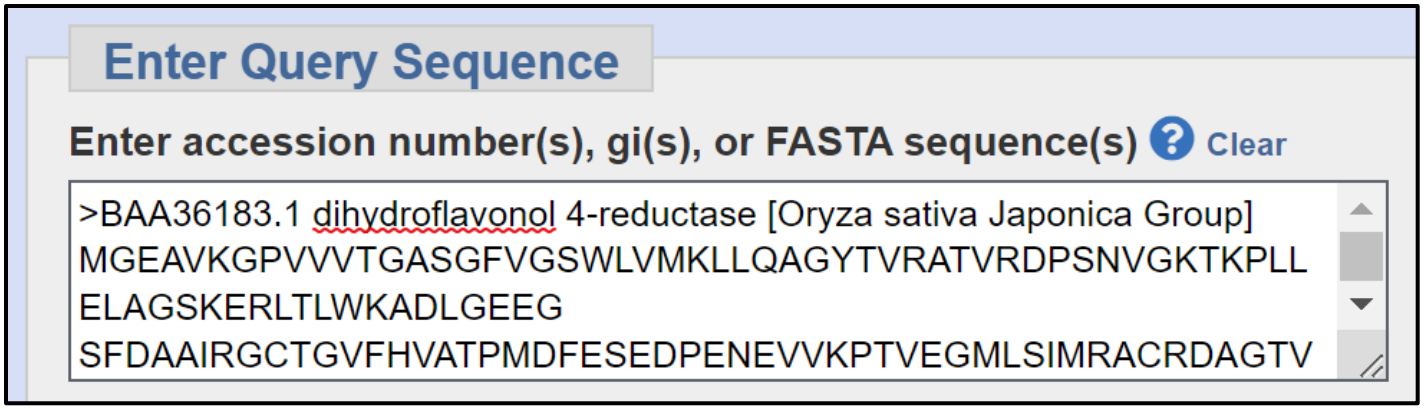
- WHY?
  - NCBI refseq_protein is a collection of translated nucleotides of coding genes in NCBI's current annotation.
  - The source nucleotide sequence of BAA36183.1 may not be considered as coding in NCBI's current annotation.
  - And it is likely that the nucleotide sequence does exists in the rice genome.

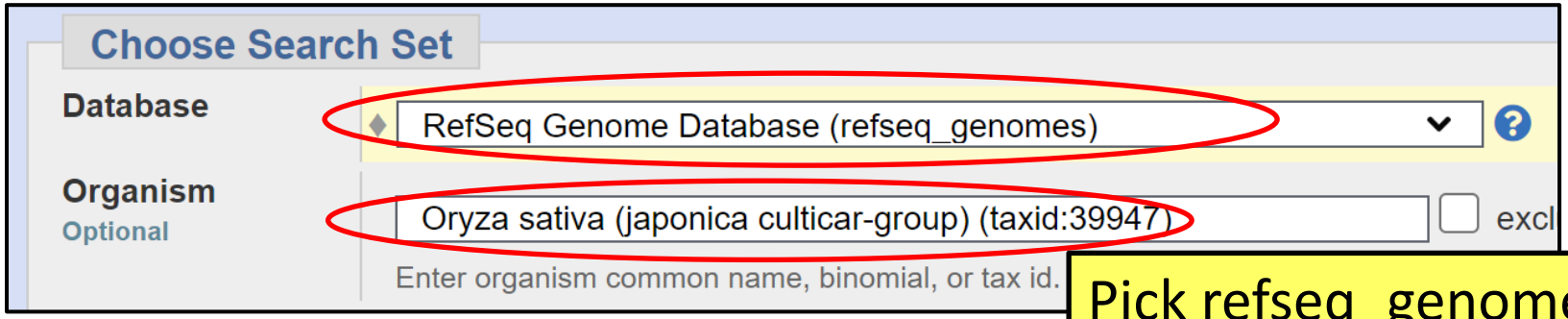
NCBI BLAST services

- How to find the source location of protein sequence BAA36183.1?
  - We have a protein query.
  - We want to search it against a genome (a set of nucleotide sequences)
  - We should apply TBLASTN

NCBI BLAST services

- In TBLASTN page





Pick refseq_genomes, and Oryza sativa japonica

NCBI BLAST services

- In this example, we are not getting an 100% identity alignment by querying this rice protein against rice proteins in the NCBI refseq_prot database.

Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments Download Select columns Show 100

select all *9 sequences selected* GenBank Graphics

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Oryza sativa Japonica Group chromosome 1, ASM3414082v1	Oryza sativa Japonica Group	383	1663	100%	3e-119	89.52%	43929697	NC_089035.1
Oryza sativa Japonica Group chromosome 8, ASM3414082v1	Oryza sativa Japonica Group	160	780	87%	6e-42	2.92%	28605474	NC_089042.1
Oryza sativa Japonica Group chromosome 9, ASM3414082v1	Oryza sativa Japonica Group	101	1567	86%	1e-31	30.12%	27474823	NC_089043.1
Oryza sativa Japonica Group chromosome 2, ASM3414082v1	Oryza sativa Japonica Group	115						36.1

Best hit sequence got 90% identity

NCBI BLAST services

- By examining the alignments, we have three pieces of near 100% identity alignments and close to each other
  - from protein to genome
  - Should mean three exon/CDS regions
  - The “90% identity” in last slide should be an average of many alignments.

Click this for checking hit genome regions

Range 1: 25963200 to 25963829 GenBank Graphics ... t Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps	Frame
383 bits(984)	3e-119	Compositional matrix adjust.	210/210(100%)	210/210(100%)	0/210(0%)	+3

Range 2: 25962447 to 25962815 GenBank Graphics ▼ Next Match ▲ Previous Match First Match

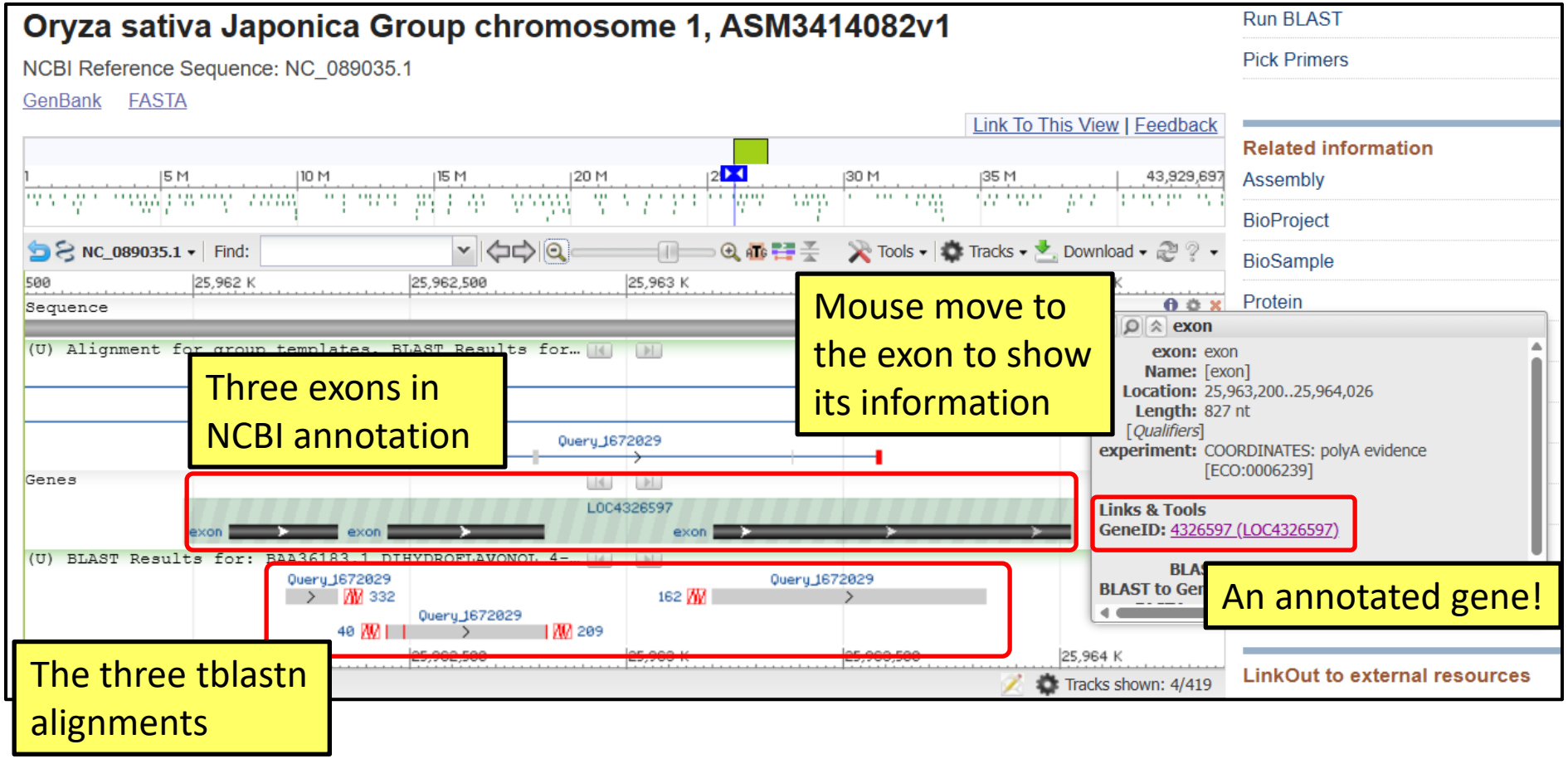
Score	Expect	Method	Identities	Positives	Gaps	Frame
253 bits(645)	2e-84	Compositional matrix adjust.	120/123(98%)	122/123(99%)	0/123(0%)	+3

Range 3: 25962215 to 25962334 GenBank Graphics ▼ Next Match ▲ Previous Match First Match

Score	Expect	Method	Identities	Positives	Gaps	Frame
82.4 bits(202)	2e-84	Compositional matrix adjust.	40/40(100%)	40/40(100%)	0/40(0%)	+2

NCBI BLAST services

- After some appropriate adjustments



NCBI BLAST services

- NCBI considers this gene is *pseudo* so it has not protein sequence => BLASTP cannot find a good hit

LOC4326597 dihydroflavonol 4-reductase-like [*Oryza sativa Japonica Group* (Japanese rice)]

Gene ID: 4326597, updated on 12-Jul-2024

Download Datasets

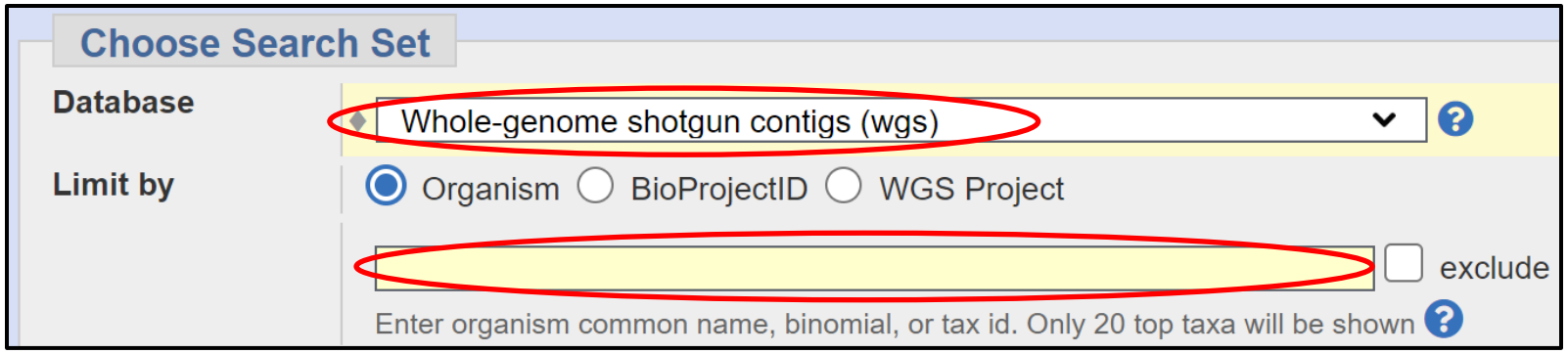
Summary

Gene symbol	LOC4326597
Gene description	dihydroflavonol 4-reductase-like
Gene type	pseudo
RefSeq status	MODEL
Organism	Oryza sativa Japonica Group
Lineage	Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta; Spermatophyta; Magnoliopsida; Liliopsida; Poales; Poaceae; BOP clade; Oryzoideae; Oryzeae; Oryzinae; Oryza; Oryza sativa

NEW **Try the new Gene table**
Try the new Transcript table

NCBI BLAST services

- A short note on searching against genomes
  - It is possible to search against *draft* genomes that were submitted to NCBI, *if available*
  - By picking “wgs” for Database and input corresponding organism
  - NCBI Taxonomy should help you to find the correct organism name.

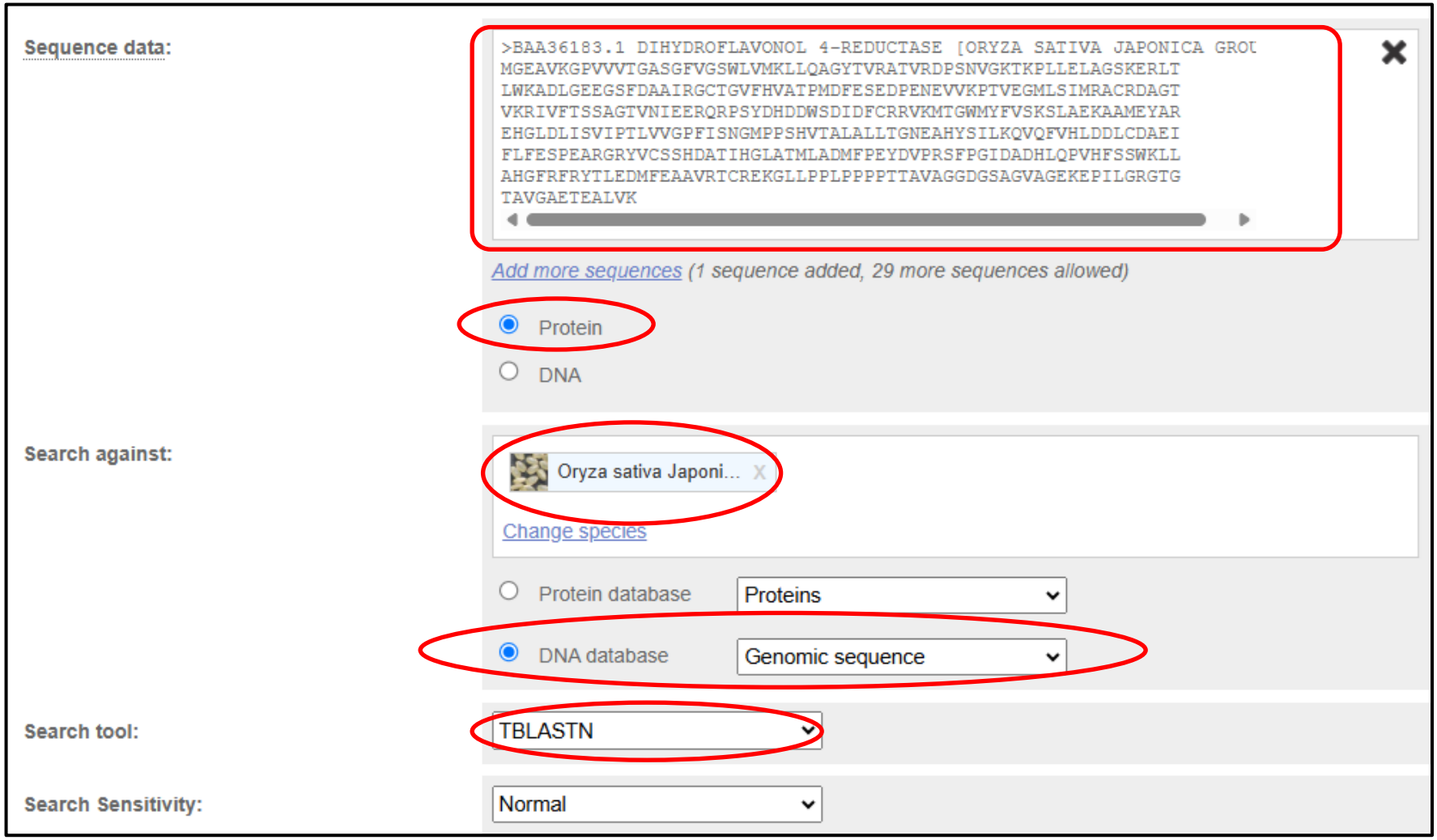


Ensembl BLAST services

- For the aim of searching protein source in the genome, Ensembl provides a better integrated interface.
- In this example, we use the same BAA36183.1 protein sequence as the query to search rice genome in the Ensembl Plants database.
  - https://plants.ensembl.org/Multi/Tools/Blast

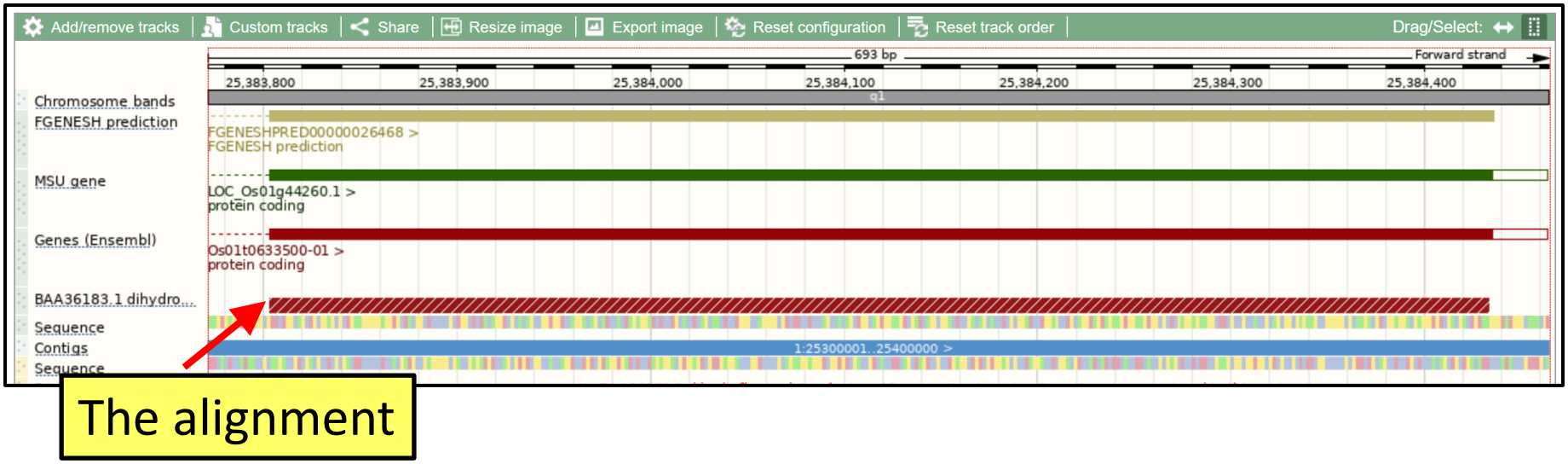
Ensembl BLAST services

- The operation should be simple.



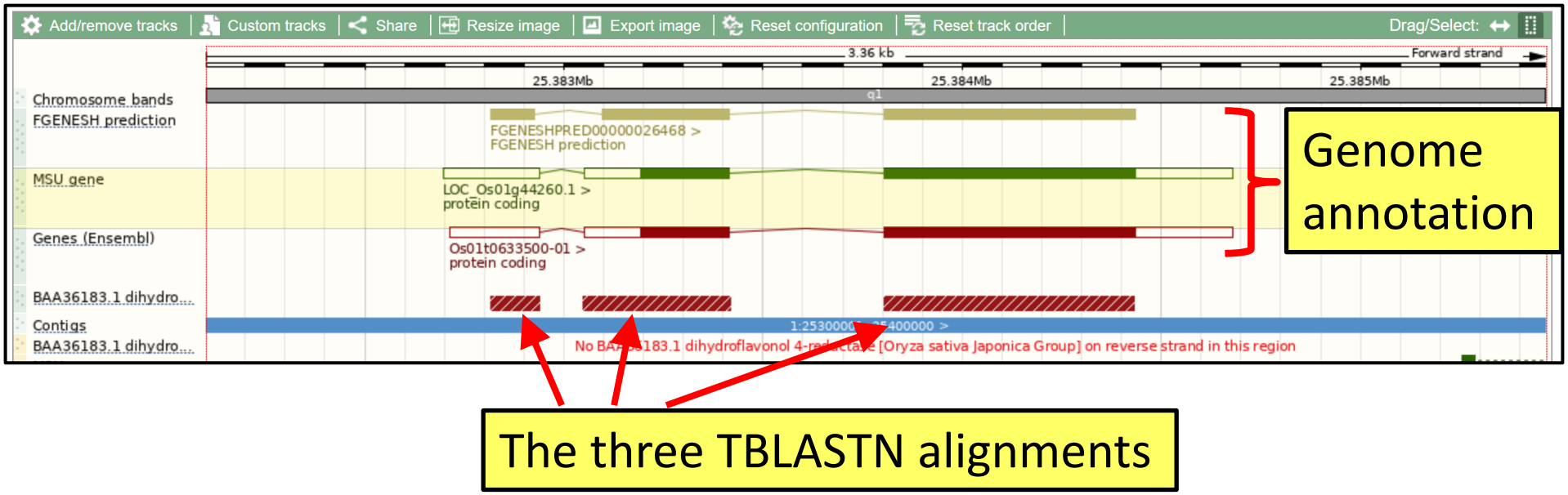
Ensembl BLAST services

- The first few alignments with %ID close to 100 are all in chromosome 1.
- Click any of it to get an integrated view of the alignments and genome annotation.



Ensembl BLAST services

- Appropriate zoom-out and zoom-in would bring us to a view of all three alignments and overlapping genome annotations



A short note on TBLASTN/TBLASTX

- TBLASTN/TBLASTX searches are very sensitive
  - the actual search is done at the protein level
  - less affected by mutations at the nucleotide level
- TBLASTN/TBLASTX would be good at
  - finding footprints of protein/nucleotide in a nucleotide database
    - Including genome database

General guide line of using BLAST

- What to do if no desired search results?
  - Set organism constraint
    - Check numbers of Nucleotide/Protein in corresponding Taxonomy page (for NCBI)
  - Loose E-value threshold
    - BLAST statistics
  - Adjust Maximum target sequences
    - BLAST reports *top* sequences but not *first* sequences during the search

General guide line of using BLAST

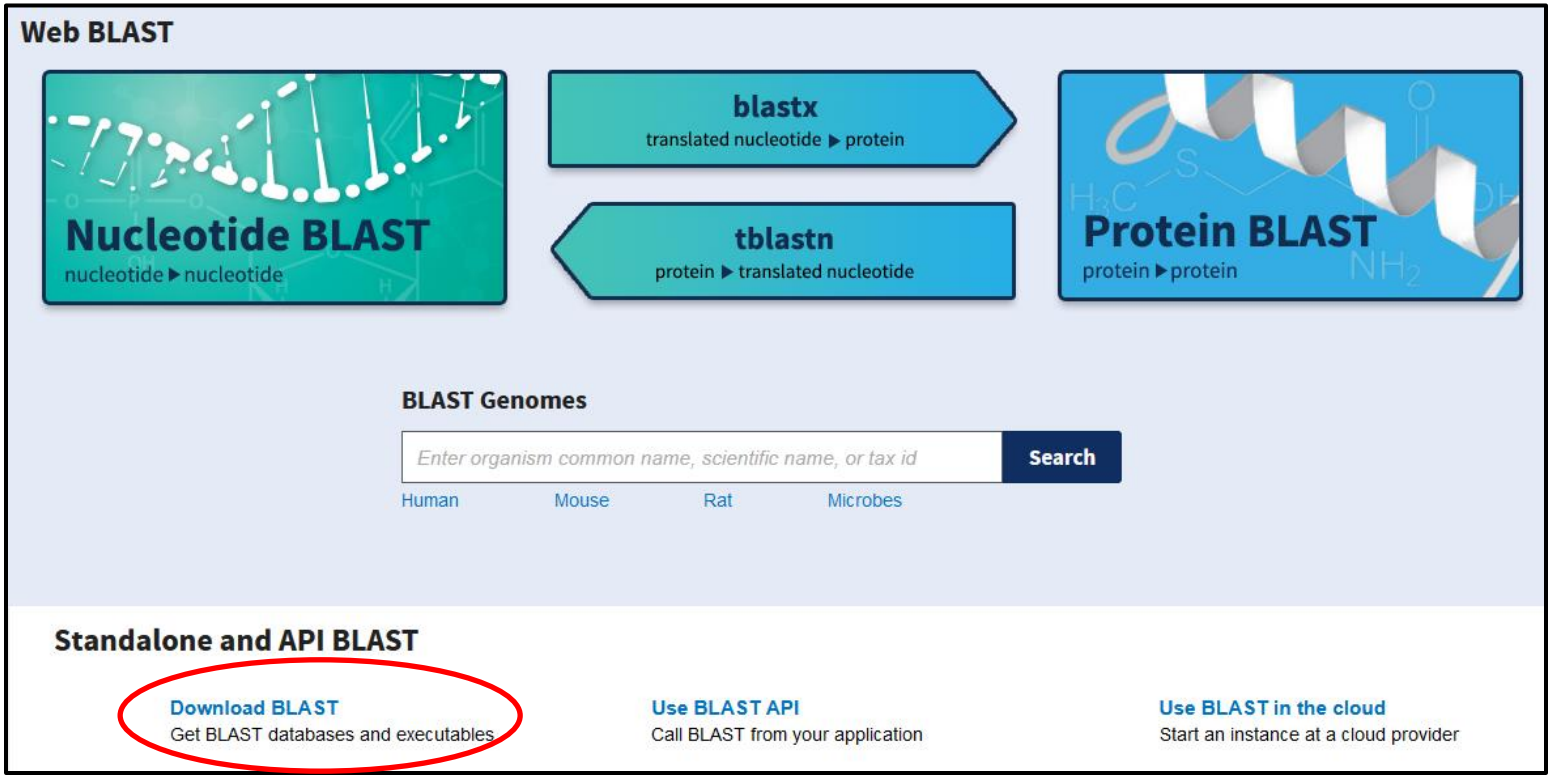
- What to do if no desired search results? (cont.)
  - Shorten Word size
    - BLAST algorithm
  - Change scoring matrix if the target is divergent from your source
    - for non-BLASTN programs
  - Choose a sensitive BLAST program
    - megablast -> blastn -> blastp -> blastx/tblastn -> tblastx

Short summaries

- NCBI BLAST services provide
  - enriched options for controlling the program and
  - many options of databases.
  - The interface is strongly integrated with the NCBI database.
- Ensembl BLAST services provide a good integration for visualization of alignments and genome annotation

Standalone BLAST programs

- The way to download BLAST executables
  - The same entry page of BLAST pages
  - Go and follow the download link inside it



Standalone BLAST programs

- The way to download BLAST executables (cont.)
  - Executables for many platforms are available
    - Windows, MacOS, and Linux

Name	Last modified	Size
Parent Directory		-
ChangeLog	2024-06-25 14:34	85
ncbi-blast-2.16.0+-1.src.rpm	2024-06-25 14:31	21M
ncbi-blast-2.16.0+-1.src.rpm.md5	2024-06-25 14:35	63
ncbi-blast-2.16.0+-1.x86_64.rpm	2024-06-25 14:31	202M
ncbi-blast-2.16.0+-1.x86_64.rpm.md5	2024-06-25 14:35	66
ncbi-blast-2.16.0+-aarch64-linux.tar.gz	2024-07-30 11:04	225M
ncbi-blast-2.16.0+-aarch64-linux.tar.gz.md5	2024-07-30 11:04	74
ncbi-blast-2.16.0+-aarch64-macosx.tar.gz	2024-06-25 14:33	191M
ncbi-blast-2.16.0+-aarch64-macosx.tar.gz.md5	2024-06-25 14:35	75
ncbi-blast-2.16.0+-aarch64.dmg	2024-06-25 14:33	193M
ncbi-blast-2.16.0+-aarch64.dmg.md5	2024-06-25 14:35	65
ncbi-blast-2.16.0+-src.tar.gz	2024-06-25 14:35	27M
ncbi-blast-2.16.0+-src.tar.gz.md5	2024-06-25 14:35	64
ncbi-blast-2.16.0+-src.zip	2024-06-25 14:35	31M
ncbi-blast-2.16.0+-src.zip.md5	2024-06-25 14:35	61
ncbi-blast-2.16.0+-universal-macosx.tar.gz	2024-06-25 14:44	398M
ncbi-blast-2.16.0+-universal-macosx.tar.gz.md5	2024-06-25 14:44	76
ncbi-blast-2.16.0+-universal.dmg	2024-06-25 14:43	400M
ncbi-blast-2.16.0+-universal.dmg.md5	2024-06-25 14:44	66
ncbi-blast-2.16.0+-win64.exe	2024-06-25 14:30	129M
ncbi-blast-2.16.0+-win64.exe.md5	2024-06-25 14:35	63
ncbi-blast-2.16.0+-x64-linux.tar.gz	2024-06-25 14:33	246M
ncbi-blast-2.16.0+-x64-linux.tar.gz.md5	2024-06-25 14:35	70
ncbi-blast-2.16.0+-x64-macosx.tar.gz	2024-06-25 14:35	206M
ncbi-blast-2.16.0+-x64-macosx.tar.gz.md5	2024-06-25 14:35	71
ncbi-blast-2.16.0+-x64-win64.tar.gz	2024-06-25 14:31	133M
ncbi-blast-2.16.0+-x64-win64.tar.gz.md5	2024-06-25 14:35	70
ncbi-blast-2.16.0+-x86_64.dmg	2024-06-25 14:34	208M
ncbi-blast-2.16.0+-x86_64.dmg.md5	2024-06-25 14:35	64

Standalone BLAST programs

- Next slides are a walkthrough under a fresh new Ubuntu24 server
  - Using ASCS: https://ascs.sinica.edu.tw/
- This walkthrough will present
  - install ncbi-blast+ from the ubuntu distribution
  - install most updated ncbi+blast+ from downloaded executables
  - performing TBLASTN/TBLASTX by querying a protein/nucleotide sequence against the rice genome for footprint searches
    - with a few small scripts for post-processing

Standalone BLAST programs

- Related files including the walkthrough log (process.txt) can be found at
  - https://maccu.project.sinica.edu.tw/20250513/
- **CAUTION**: Better *not* copy command from this PowerPoint file. Office might twist symbols like - ‘ “.
  - Please refer the walkthrough log process.txt at the above URL.

Standalone BLAST programs

- 1. install ncbi blast+ from ubuntu distribution packages

```
ubuntu@blast:~$ sudo apt update

ubuntu@blast:~$ sudo apt install ncbi-blast+
(...)
Setting up ncbi-blast+ (2.12.0+ds-4build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...

ubuntu@blast:~$ blastn -version
blastn: 2.12.0+
 Package: blast 2.9.0, build Sep 30 2019 01:57:31
```

Standalone BLAST programs

- 2. (optional) install most-updated ncbi blast+ programs

```
ubuntu@blast:~$ curl -O
https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/n
cbi-blast-2.16.0+-x64-linux.tar.gz

ubuntu@blast:~$ tar -zxvf ncbi-blast-2.16.0+-x64-linux.tar.gz

ubuntu@blast:~$ tail -n 2 ~/.bashrc
PATH="/home/ubuntu/ncbi-blast-2.16.0+/bin:$PATH"
export PATH

ubuntu@blast:~$ source ~/.bashrc

ubuntu@blast:~$ blastn -version
blastn: 2.16.0+
 Package: blast 2.16.0, build Jun 25 2024 08:58:03
```

Standalone BLAST programs

- 3. Download genome files

```
(download genome FASTA file)
ubuntu@blast:~$ wget http://ftp.ensemblgenomes.org/pub/plants/release-
61/fasta/oryza_sativa/dna_index/Oryza_sativa.IRGSP-
1.0.dna.toplevel.fa.gz

(download genome annotation GFF3 files)
ubuntu@blast:~$ wget http://ftp.ensemblgenomes.org/pub/plants/release-
61/gff3/oryza_sativa/Oryza_sativa.IRGSP-1.0.61.gff3.gz

(unzip files)
ubuntu@blast:~$ gzip -d Oryza_sativa.IRGSP-1.0.dna.toplevel.fa.gz
ubuntu@blast:~$ gzip -d Oryza_sativa.IRGSP-1.0.61.gff3.gz
```

Standalone BLAST programs

- 4. Install necessary programs

```
ubuntu@blast:~$ wget
https://maccu.project.sinica.edu.tw/20250513/ExampleData.tar.gz
ubuntu@blast:~$ tar -zxvf ExampleData.tar.gz

ubuntu@blast:~$ wget
https://downloads.sourceforge.net/project/rackj/0.99b/rackJ.tar.gz
ubuntu@blast:~$ tar -zxvf rackJ.tar.gz

ubuntu@blast:~$ sudo apt install default-jre
ubuntu@blast:~$ sudo apt install bioperl

ubuntu@blast:~$ tail -n 3 ~/.bashrc
PATH="/home/ubuntu/ncbi-blast-2.16.0+/bin:$PATH"
PATH="/home/ubuntu/rackJ/scripts:$PATH"
export PATH

ubuntu@blast:~$ source ~/.bashrc
```

Standalone BLAST programs

- 5. TBLASTN BAA36183.1 against rice genome and corresponding post processing

```
ubuntu@blast:~$ tblastn -subject Oryza_sativa.IRGSP-1.0.dna.toplevel.fa -query
ExampleData/BAA36183.1.fasta -outfmt "7 qaccver qlen sallacc slen pident length
nident positive gapopen qstart qend sstart send sframe evalue bitscore" -out
BAA36183.1.tblastn.txt -word_size 3 -window_size 0 -evalue 10

(create tmp file for consistentIterator input)
ubuntu@blast:~$ cat BAA36183.1.tblastn.txt | perl -ne 'chomp; next if /^#/; chomp;
@t=split; if($t[11]>$t[12]){ print join("\t",@t[2,0,12,11,10,9])."\t$_\n" }else{
print join("\t",@t[2,0,11,12,9,10])."\t$_\n" }' > BAA36183.1.tblastn.tmp

(maximum gene size in chromosomes)
ubuntu@blast:~$ cat Oryza_sativa.IRGSP-1.0.61.gff3 | perl -ne 'if(/^#/){}else{
@t=split(/\t/); $len=$t[4]-$t[3]+1; $max=$len if $len>$max && $t[2] eq "gene"}
print "$max\n" if eof'
57648

(consistentIterator)
ubuntu@blast:~$ consistentIterator.pl -parapass "-min 20 -max 20 -score 22 -
limitRef 57648 -refKeep -queryKeep -strandKeep -order" BAA36183.1.tblastn.tmp
/home/ubuntu/rackJ/rackj.jar | cut -f 1,8- > BAA36183.1.tblastn.grp.xls
```

Important parameters of BLAST

- -outfmt <format_string>
  - specify the output format
  - be sure to apply “-help” to get detailed information
  - In our tblastn example, we applied
    - -outfmt "7 qaccver qlen sallacc slen pident length nident positive gapopen qstart qend sstart send sframe evalue bitscore“
    - which means text tabular output with specified information as columns (query accession, query length, subject accession, subject length, …)

Important parameters of BLAST

- -evalue <real_number>: E-value cutoff
  - to filter out alignments with E-values larger than the cutoff
- -dust (BLASTN) OR -seg (non-BLASTN)
  - to filter low complexity regions
  - the default setting might be different from one BLAST program to the other, apply -help to check it.
- -num_threads <integer>
  - number of processors to use
    - would *speed up* BLAST search for multi-core CPU

Important parameters of BLAST

- -matrix <matrix string> (default BLOSUM62)
  - specify scoring matrix for protein alignments

Matrix	Best use	Similarity (%)
BLOSUM90	Short alignments that are highly similar	70-90
BLOSUM80	Detecting members of a protein family	50-60
BLOSUM62	Most effective in finding all potential similarities	30-40
BLOSUM30	Longer alignments of more divergent sequences	<30

Important parameters of BLAST

- -word_size <integer>: index word size (for the look-up table)
  - Increasing this parameter would increase search speed at a price of sensitivity.
- -window_size <integer>:
  - Set this to 0 to apply 1-hit algorithm to increase sensitivity at a cost of search speed.
  - 1-hit may be needed for short query sequence.

Standalone BLAST programs

- 6. (optional) TBLASTX AB003496.1 (source nucleotide of BAA36183.1) against rice genome and corresponding post processing

```
ubuntu@blast:~$ tblastx -subject Oryza_sativa.IRGSP-1.0.dna.toplevel.fa
-query ExampleData/AB003496.1.fasta -out AB003496.1.tblastx.out -evalue
10

ubuntu@blast:~$ ./ExampleData/parseTBLASTX.pl AB003496.1.tblastx.out >
AB003496.1.tblastx.txt

(create tmp file for consistentIterator input)
ubuntu@blast:~$ cat AB003496.1.tblastx.txt | perl -ne 'chomp; next if
/^#/; chomp; @t=split; if($t[11]>$t[12]){ print
join("\t",@t[2,0,12,11,10,9])."\t$_\n" }else{ print
join("\t",@t[2,0,11,12,9,10])."\t$_\n" }' > AB003496.1.tblastx.tmp

(consistentIterator)
ubuntu@blast:~$ consistentIterator.pl -parapass "-min 20 -max 20 -score
22 -limitRef 57648 -refKeep -queryKeep -strandKeep -order"
AB003496.1.tblastx.tmp /home/ubuntu/rackJ/rackj.jar | cut -f 1,8- >
AB003496.1.tblastx.grp.xls
```

Standalone BLAST programs

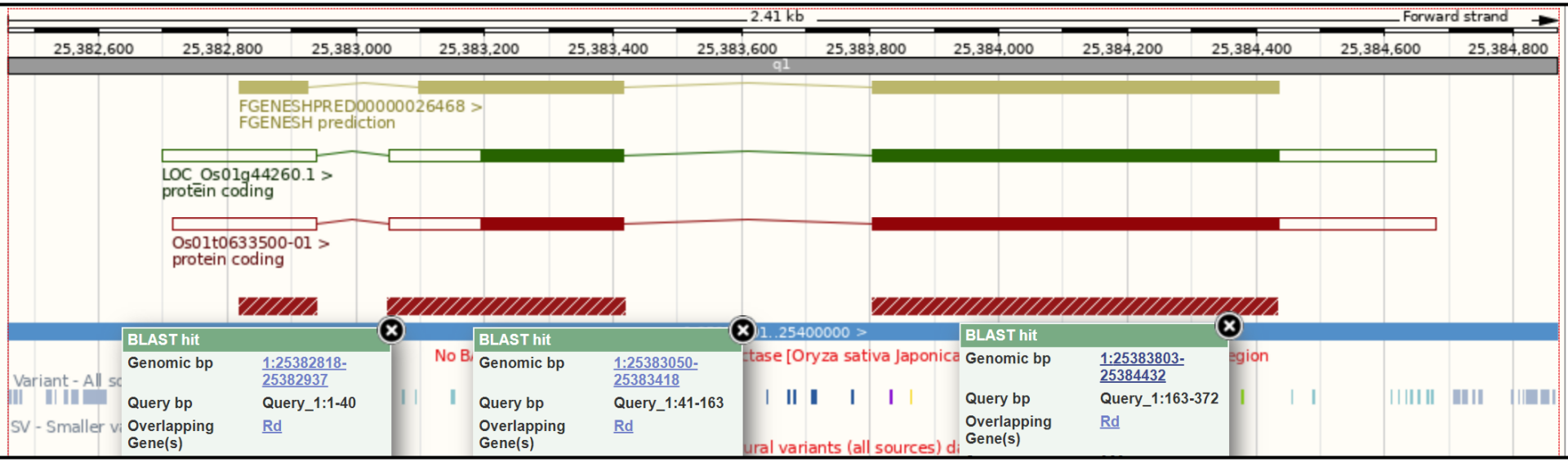
- The use of the consistentIterator.pl script is to partition fragmented BLAST alignments into *co-linear and non-overlapping* groups.
  - We saved the grouped TBLASTN and TBLASTX results in BAA36183.1.footprint.xlsx.

alnGroup	chr	identity	aln length	qstart	qend	sstart	send	frame	evalue	bitscore
1	1	100	40	1	40	25382818	25382937	1	1.74E-84	82.4
1	1	97.561	123	41	163	25383050	25383418	2	1.74E-84	253
1	1	100	210	163	372	25383803	25384432	2	2.58E-119	383

1st group of alignments in sheet BAA36183.1.tblastn.grp of Excel file BAA36183.1.footprint.xlsx

Standalone BLAST programs

- This 1st *co-linear and non-overlapping* alignment group visualized in the Ensembl website.
  - Our query is 372 AA's.



Query 1-40

Query 41-163

Query 163-372

Short summaries

- Standalone BLAST programs give us abilities to
  - programmatically run BLAST programs,
  - designate BLAST output information, and
  - postprocessing the outputs.
- The consistentIterator.pl script can be used to
  - group low similarity and fragmented BLAST alignments into *co-linear and non-overlapping* groups.

Short summaries

- Grouped alignments with high coverage to the query usually means a confident footprint.
- In practice, we ever see 30 alignments with bitscore<100 (low similarity) been grouped together for a 2500bp TBLASTX query.
  - Highly mutated footprint.

Finally

- Thank you for your attentions.
- I am willing to answer and/or discuss questions via email or in some other interactive form.
  - Please don’t hesitate to let me know if you have any questions.