Group3

Identifying Proteins via K-mers Assembly

Introduction

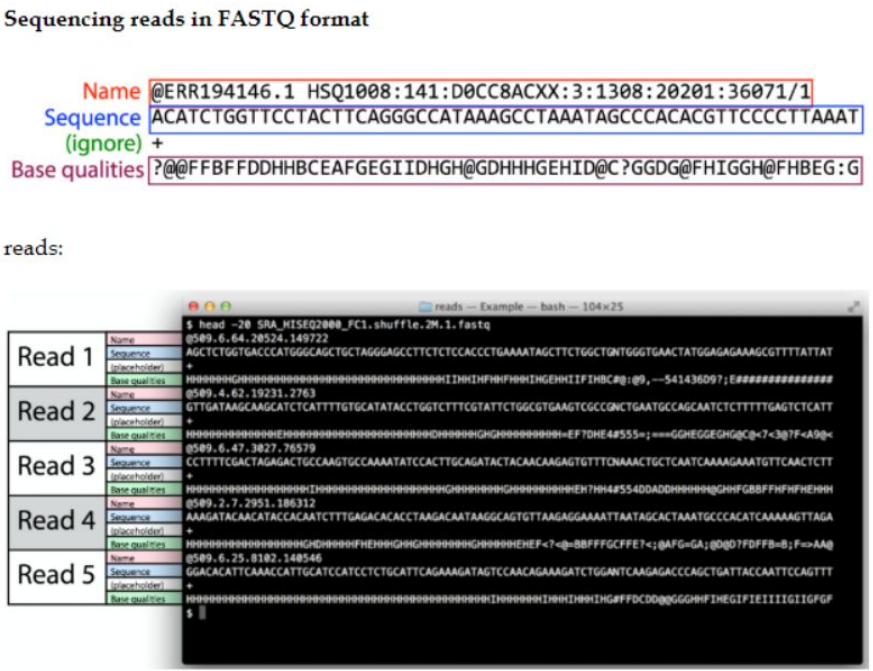
- Predicting proteins from DNA sequences can provide valuable insights into their roles in biological processes.
- Proteins perform a variety of essential functions
- Predicting proteins can help in identifying potential targets for drug development and in understanding the evolution and relationships between different organisms.
- Upload a DNA sequence file and obtain a list of predicted proteins.
- By parsing the DNA file, performing de novo sequence K-mers assembly using optimized algorithms, and utilizing the NCBI BLAST REST API to predict the proteins.

Introduction

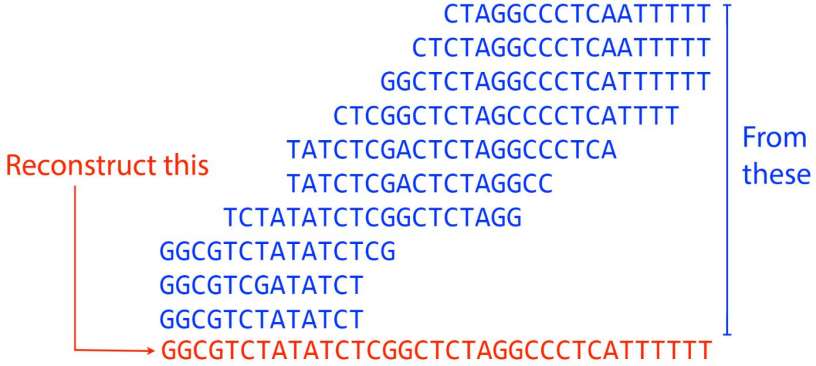
- Parsing FASTA /FASTQ file:
  - Read the FASTA/FASTQ file containing DNA sequences
  - Extract the relevant sequence(s) for analysis
- Alignment with SCS algorithm:
  - Use the suitable algorithm to align the DNA sequences to get shortest common superstring
- Transcription and translation:
  - Transcribe the DNA sequence(s) into mRNA
  - Translate the mRNA sequence(s) into amino acid sequences
- Send as query to BLAST:
  - Use BLAST to search for known proteins that match the amino acid sequence(s)

Parsing - Fastq and Fasta Files

- Include quality information for each base.
- ASCII - encoded (using character to encode an integer)



De Novo DNA Assembly

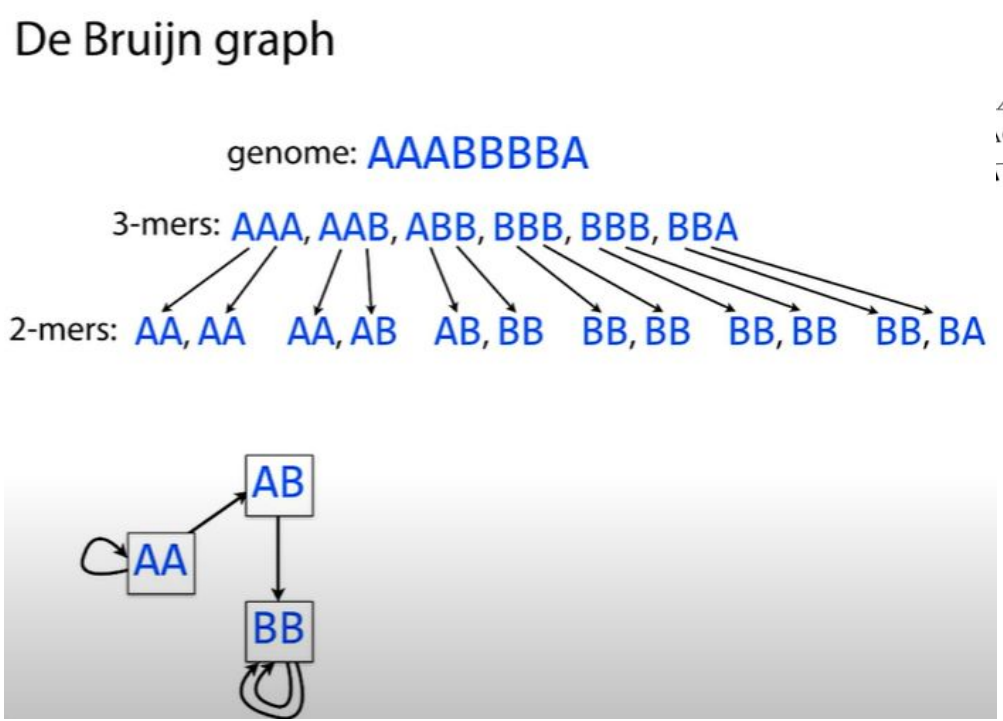


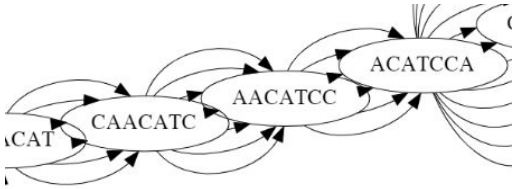
- Shortest Common Superstring (SCS)
  - NP-Complete
  - For different order we get different SCS
- Shortcoming repetitive regions tend to be collapsed

We do not consider

More things to consider:
- mismatches
- indels

Alternative for SCS





- Directed graph
- which avoids overlapping problem somehow.
- All nodes corresponds to distinct k-1-mers from the genome.

SCS and Greedy SCS

SCS

Idea: pick order for strings in *S and* construct superstring

order 1: AAA AAB ABA ABB BAA BAB BBA BBB

AAABABBAABABBABBB ⟵ superstring 1

order 2: AAA AAB ABA BAB ABB BBB BAA BBA

AAABABBBAABBA ⟵ superstring 2

Try all possible orderings and pick shortest superstring

If *S* contains *n* strings, *n* ! (*n* factorial) orderings possible

Greedy SCS

1. Calculate overlap length for each pair of strings.
2. Find 2 strings with the biggest overlap lengths and merge them into one.
3. Repeat 2. until only one string is left.

\+ Works much faster.

\- But correct answer is not guaranteed.

Maybe it is even better!

Greedy SCS: Running time

For dataset of n=1870 strings (***~10min***):

- Pairwise overlap computation: ***80s***.
- Greedy iterations: ***513s***.

Speed up options:

- Pairwise overlap computation: ***parallelize***
- Greedy iterations: ***cannot parallelize, make long operations faster***

Longest steps during greedy iteration (**~0.5s**):

- Finding max overlap (**~0.3s**)
- Calculating overlaps between new string and old ones (**~0.2s**)

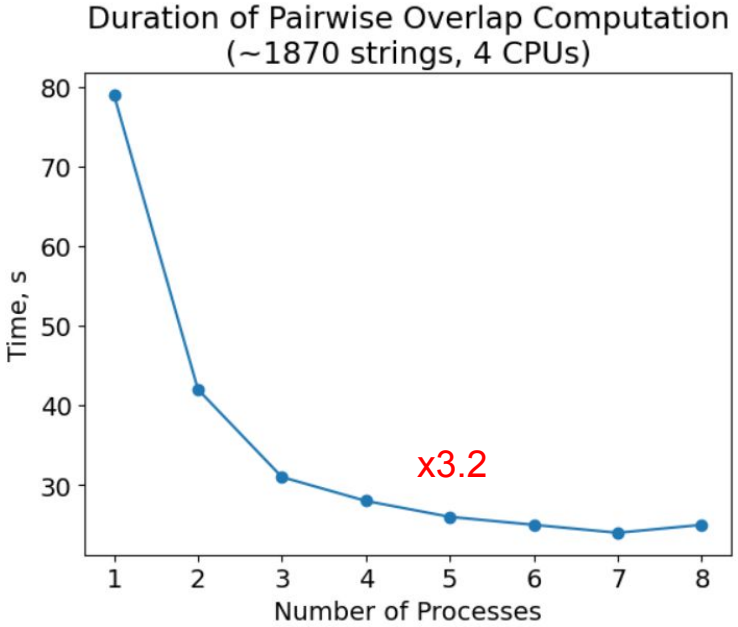
- Finding max overlap: ***use other data structure***
- Calculating overlaps: **nothing** (*parallelization might work slower)*

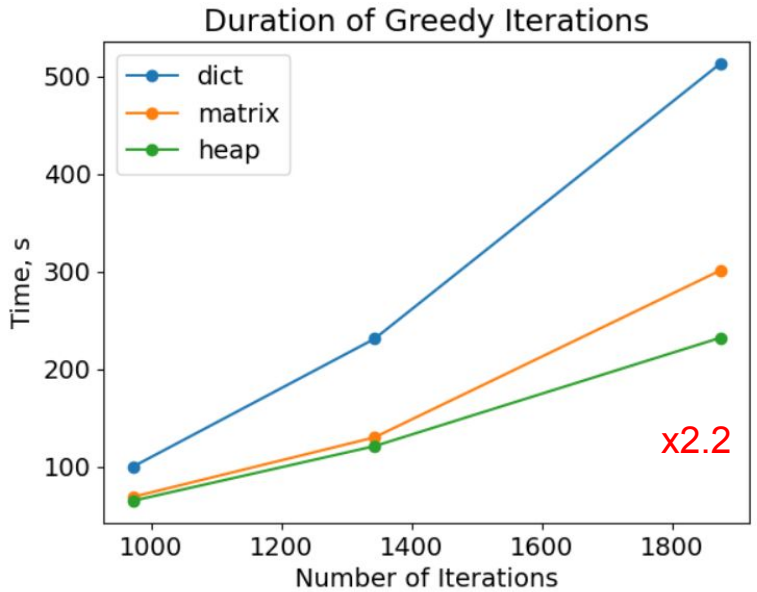
Complexities of structures for storing overlap lengths
N ~ n x n

	dict	**matrix (numpy)**	**max-heap (lazy deletion)**
get_max	N	N	log(N)*k
add	1	N	log(N)
delete	1	N	1

+ heap construction time: O(N)

Greedy SCS: Time Measurement



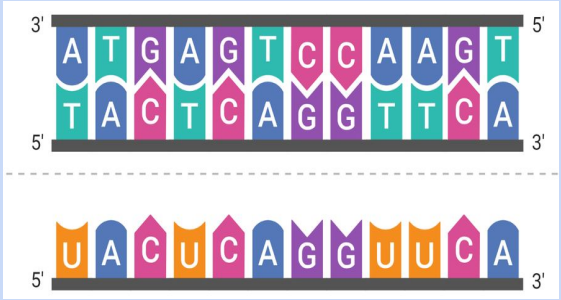


Total Time (1870 reads):

Dict + 1 Process:	592s (9min 52s)
Heap + 4 Processes:	257s (4min 17s)

x2.3

Transcription

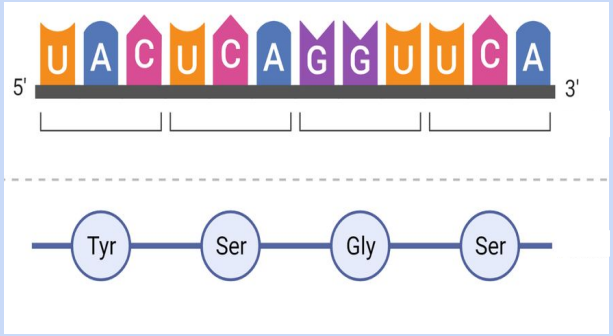


DNA AAGGCCCTCTAAGGCCCTCT from the fastq file

RNA UUCCGGGAGAUUCCGGGAGA transcription “like in a cell”

AAGGCCCUCUAAGGCCCUCU transcription of complementary strand

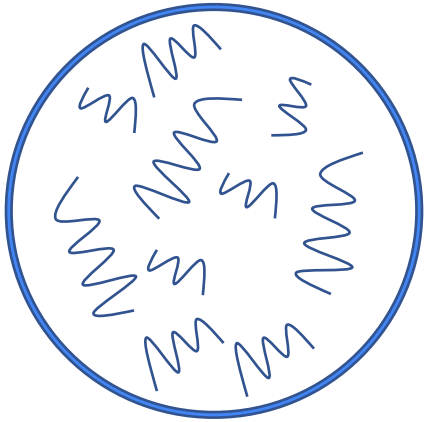
Translation



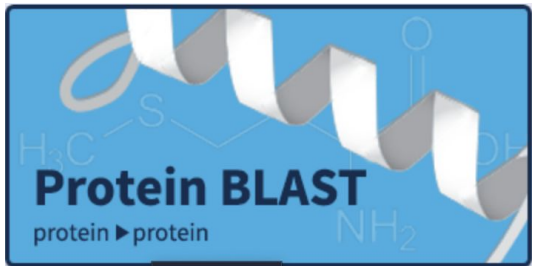
RNA UAUGAAUAUCAAUGCUUGA Product of Transcription function

Protein Asn-Ile-Asn-Ala Amino acid sequence

Database query



Possible proteins



Found in database

Challenges with Database querying

We needed …

… a database to find a protein for an amino acid sequence

… programmatic access to the database

… the functionality to download the information (json, xml)

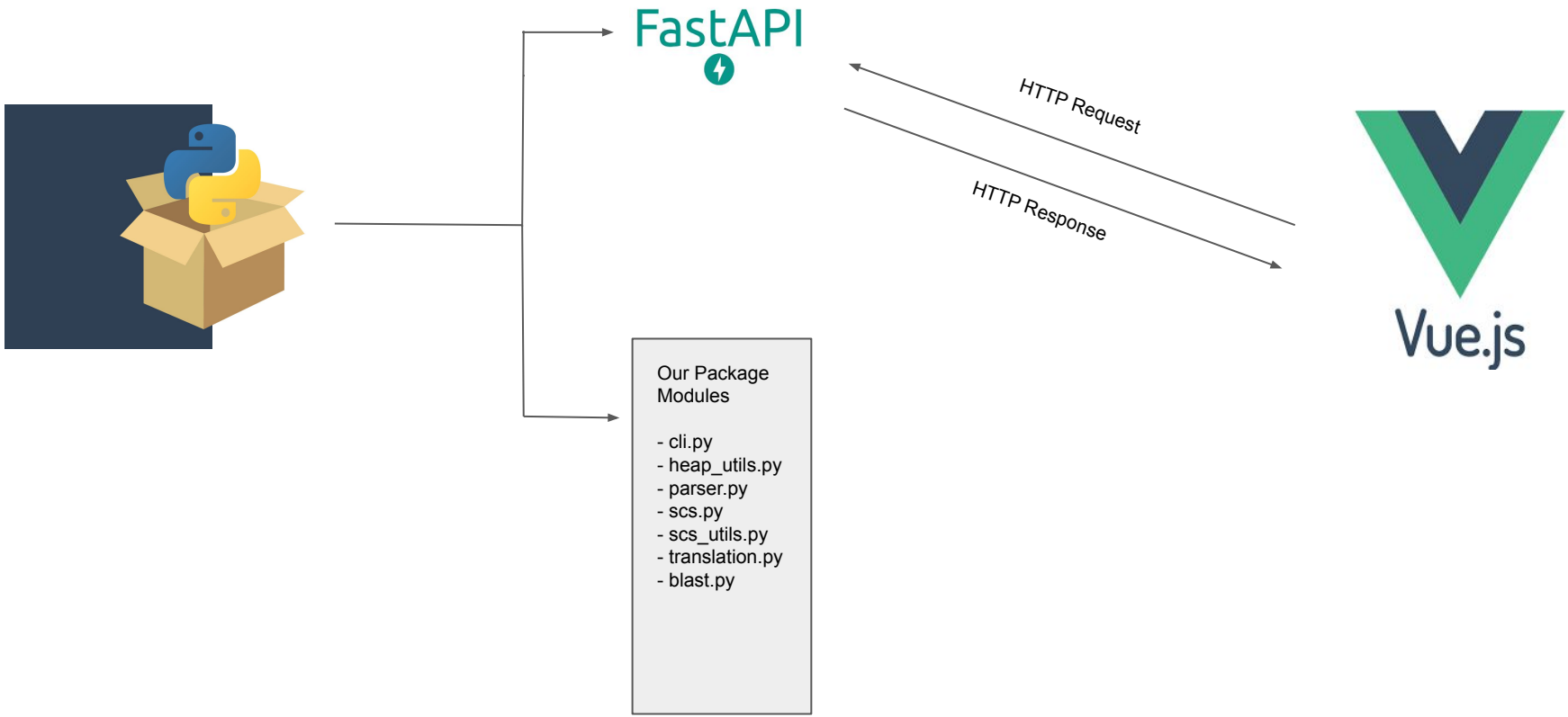
… a stable and fast connection

Challenges with Database querying

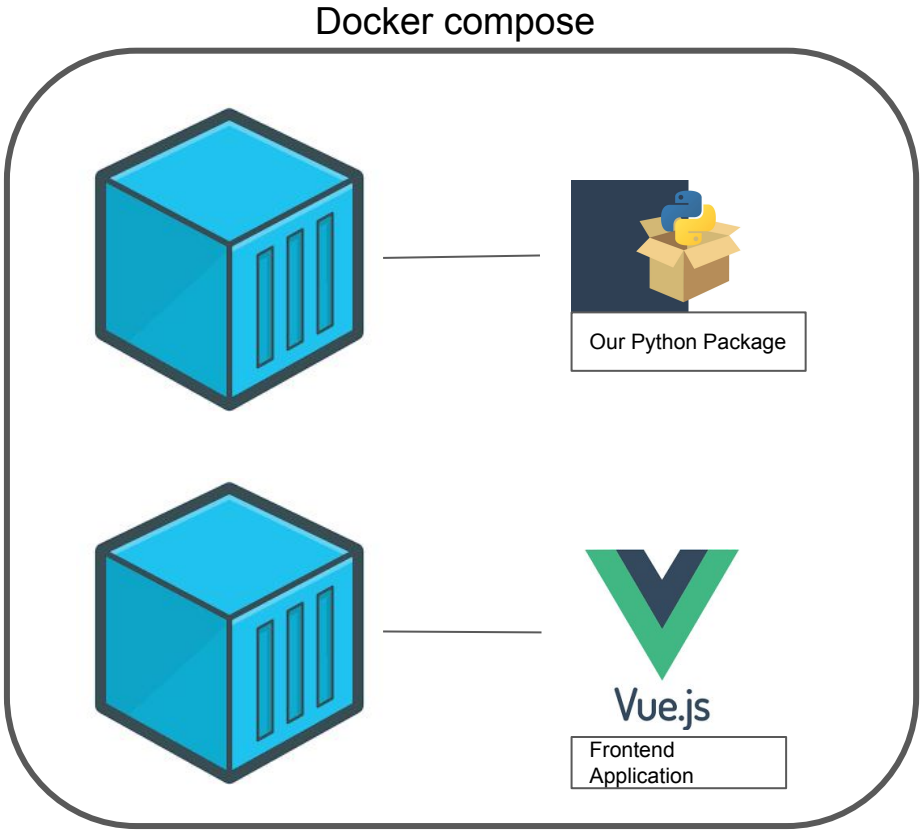
BLAST requests can take a lot of time (about 10 minutes per sequence)

- query with multiple sequences at once
- parse the BLAST website directly

Design


FastAPI
HTTP Request
HTTP Response
Vue.js
Our Package Modules
- cli.py
- heap_utils.py
- parser.py
- scs.py
- scs_utils.py
- translation.py
- blast.py

Containerization



Challenges:

- Stability of docker containers when allocating more CPU's from host machine for parallelization
- Unexpected installation errors while installing dependencies via npm for Vue.js dependent on host machine OS
- Difficulty to run a browser as another service

Live Demonstration

http://localhost:8080/

Generate mRNA Sequence

Hide Table

Protein prediction is powered by the Biopython NCBIWWW Blast API. (Biopython, 'Biopython NCBIWWW Blast API,' biopython.org/docs/1.75/api/Bio.Blast.NCBIWWW.html), accessed 2/13/2023, 9:35:57 PM

Download Table

Amino Acid	Predicted Proteins
GNQAETHAWFTSRFLYHVAGFLLPMLVMGWCYAATTQVSPVKRSGAVDLVVTGLPEDVCVCGAAQPEKHRVPCEPTAEAGCKGFLETSSLSFLPLERGLVSGLHSGLDHEGAKPWQERNPTEGRKELSSREDSTRLPYMRTQKHKLTYGVQPSSTYKAQPPARLPSHKL	C-X-C chemokine receptor type 5 isoform X2 [Macaca mulatta]
LVMGWCYAATTQVSPVKRSGAVDLVVTGLPEDVCVCGAAQPEKHRVPCEPTAEAGCKGFLETSSLSFLPLERGLVSGLHSGLDHEGAKPWQERNPTEGRKELSSREDSTRLPYMRTQKHKLTYGVQPSSTYKAQPPARLPSHKL	FAD-dependent monooxygenase [Burkholderiales bacterium]
GWCYAATTQVSPVKRSGAVDLVVTGLPEDVCVCGAAQPEKHRVPCEPTAEAGCKGFLETSSLSFLPLERGLVSGLHSGLDHEGAKPWQERNPTEGRKELSSREDSTRLPYMRTQKHKLTYGVQPSSTYKAQPPARLPSHKL	FAD-dependent monooxygenase [Burkholderiales bacterium]
VFRIPDSSASAQSYQLSCPKLTLRYCLTPSCSPGWSGLCDAVLVKWVCSVTVLEIAPNRGRALKIQHTHKPAQKRSALFTAPAGEMGWGGELLGWLGQQSGHPCPDQPNWAGA	galactokinase [Mycobacterium persicum] >gbl|ORB46862.1| galactokinase [Mycobacterium persicum]
SSRVVGKNISNPGDGFKRLPHTPPALGAGSLLLSCQSPCLQGQCPLRRGEGFLLSFFFCVTSIKTQKDGVRGTDLHPHPQPSLICPLELEDQLEPLKMSTSLRQGLGGDGERKRRMGRRELPETSRARLEPRERKAAQRS	SCAI isoform X2 [Brachionus plicatilis]
EGRVGWPVAWVGGEGTERAEERPAGSRVWPSDRLDNTCKLNGSLPVAITTTPCHVAPSPKRRWGARWGSLKPKDLLQRLGVAFHVKEWAHRSRKWSACSFKLTTSPDPVGHLSNQLETRRVCYPVEKKKKVMSCEAGPFPKLKSCDSSRQGCFWASSRGKGGSPPAPRNQVKSPRGTPADKYLLARRKTQAPPKPPLNLVLRRQEQKGLGET	uncharacterized protein LOC101793105 [Anas platyrhynchos]
LLSRVCVCTRASGFRSWWGWDPWRDGETLSLLKIQKNYLGVVADAWNPKGKDTKQVKTEGKWAGAEKPRGRPHWGNNSCGSRESVFTCAPSGHGRSGASSLCITAASARPGPPGQEG	uncharacterized protein H6S33_005915 [Morchella sextelata] >gbl|KAH0614029.1| hypothetical protein H6S33_005915 [Morchella sextelata]