Mercury BLASTN: Fast Streaming DNA Sequence Comparison

Jeremy Buhler[*], Joe Lancaster[*], Arpith Jacob[*], and Roger Chamberlain[*†]

[*]Washington University in St. Louis

[†]BECS Technology, Inc.

Supported by NIH award 1-R42-HG003225-01 and NSF awards CCF-0427794 and DBI-0237902
Dr. Chamberlain is a principal of BECS Technology, Inc.

The Big Idea

- DNA sequence comparison: target for high-performance computing
- BLASTN is the standard s/w solution
- Our FPGA impl delivers comparable results in less time on realistic analyses

Overview

- **Background** and Motivation
- **Methods**: Mercury BLASTN
- **Results**: end-to-end performance
- **Perspective**: opportunities for streaming computation on biosequences

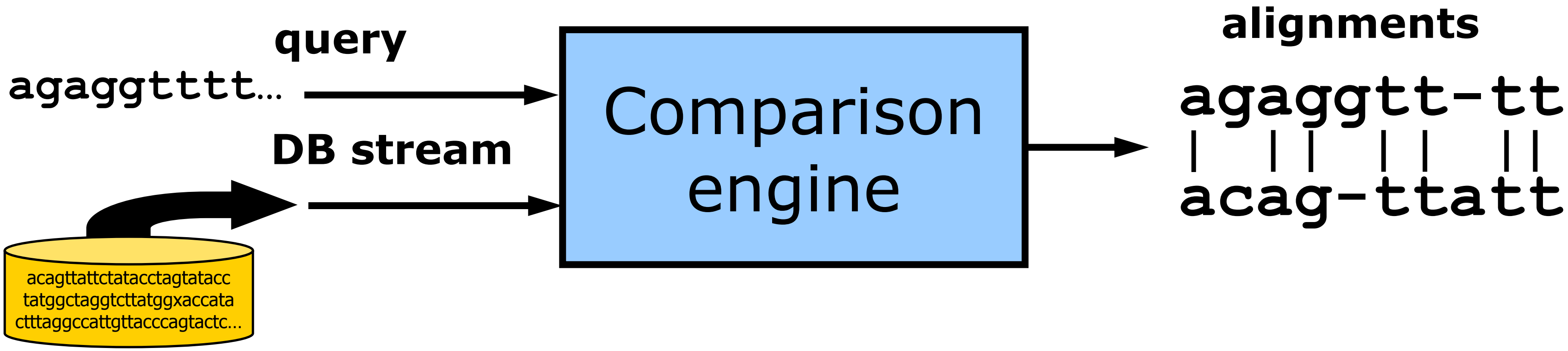
Application Goal

- Discover similarity between (parts of) two DNA sequences

```
…agaggtttt-attgcatgattcta--cta…
 |  |   ||  |     ||||||   |||
…actgaaattg-tgtacagattctccacta…
```

- **Why?** Evidence of common ancestry, perhaps similar biological function

Overview of Comparison Task



- Input
  - Query sequence: 10^2 - 10^9 DNA bases
  - Database stream: 10^9 - 10^{11} bases
- Output
  - alignments of similar substrings in query/db

Measuring Sequence Similarity

- Classical algorithm is Smith-Waterman (DP edit distance computation)

- High cost of S-W led to development of faster heuristics for searching an entire database, most notably…

Basic **L**ocal **A**lignment **S**earch **T**ool

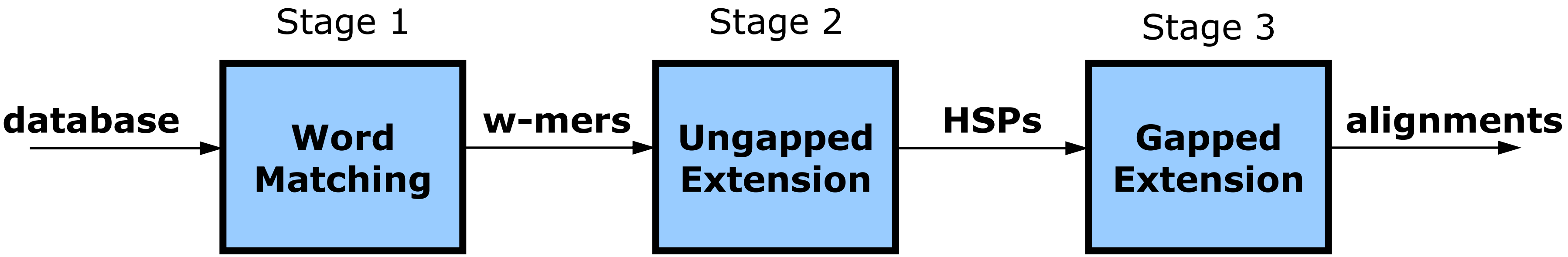
[A et al. ’90, AG ’96, A et al. ’98]

Quantifying BLAST’s Advantage

Time to compare human vs mouse genomes
(~1.5 billion bases each after prefiltering)

Smith-Waterman Software (on one modern x86 core)	~500 years
Smith-Waterman Hardware (fastest published FPGA impls)	~5 years
NCBI BLASTN Software (on one modern x86 core)	~10 days

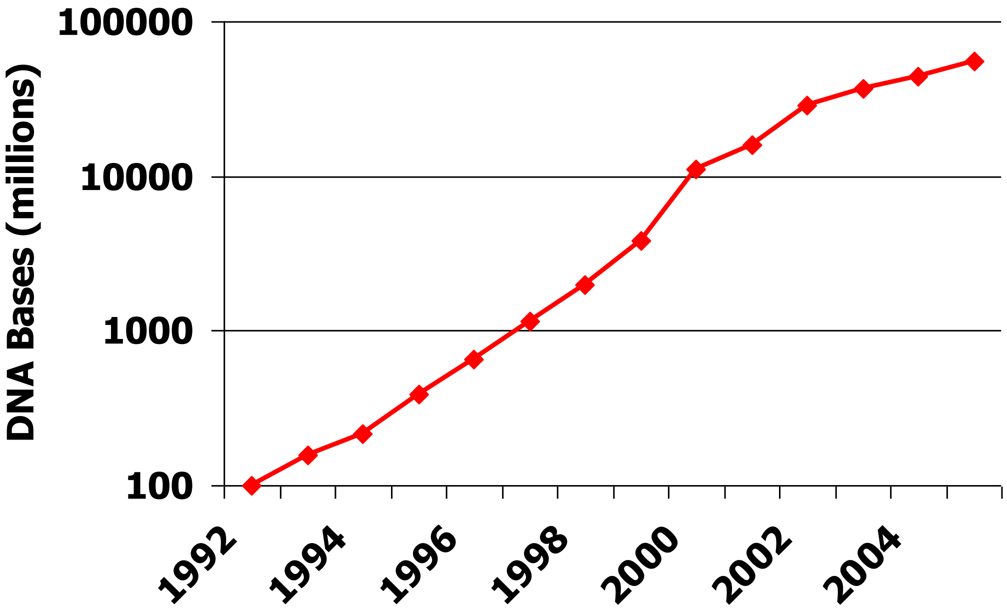
The BLASTN Filter Pipeline



DATA
COST
alignment:
HSP:
w-mer:
tcttg
Database: agtgagactcttgaat

Why Build a Faster BLAST?

Growth of NCBI GenBank



Source: NCBI

- Databases are growing exponentially
- Comparisons involve more genomes (e.g. UCSC human vs 28 species)

How to Accelerate BLAST

- Use many commodity CPUs in parallel [e.g. mpiBLAST, bglBLAST]

- Use pipeline of specialized processors
  - less hardware for same performance
  - less power, less heat
  - smaller footprint, lower maintenance

Our Contributions

- **Mercury BLAST**: high performance streaming architecture for BLASTN (and BLASTP)

- Fully implemented as FPGA/software codesign

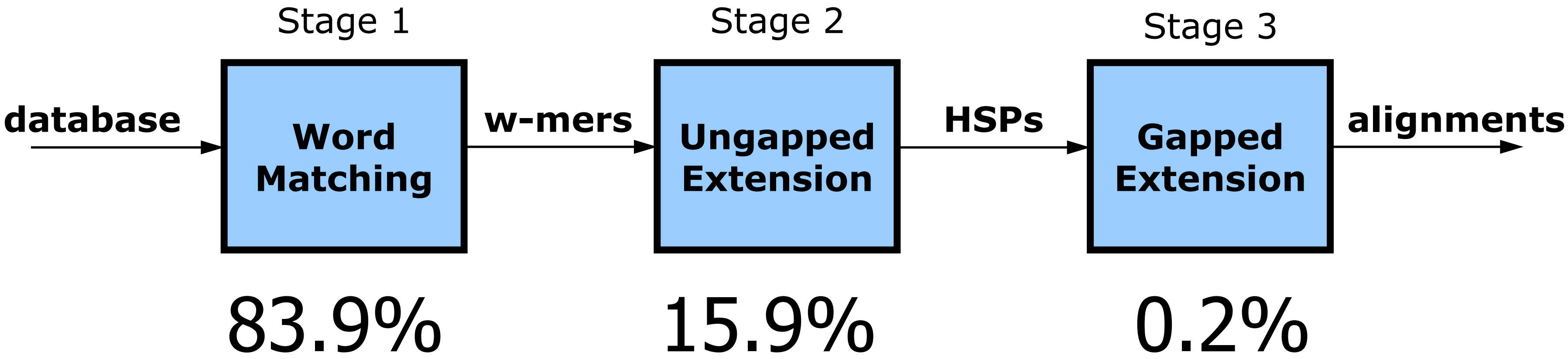
- End-to-end tests of *both* speed *and* accuracy vs NCBI BLASTN software

Overview

- **Background** and Motivation
- **Methods**: Mercury BLASTN
- **Results**: end-to-end performance
- **Perspective**: opportunities for streaming computation on biosequences

Hardware/Software Division

Software Execution Time Profile



Stage 1

Stage 2

Stage 3

database

Word Matching

w-mers

Ungapped Extension

HSPs

Gapped Extension

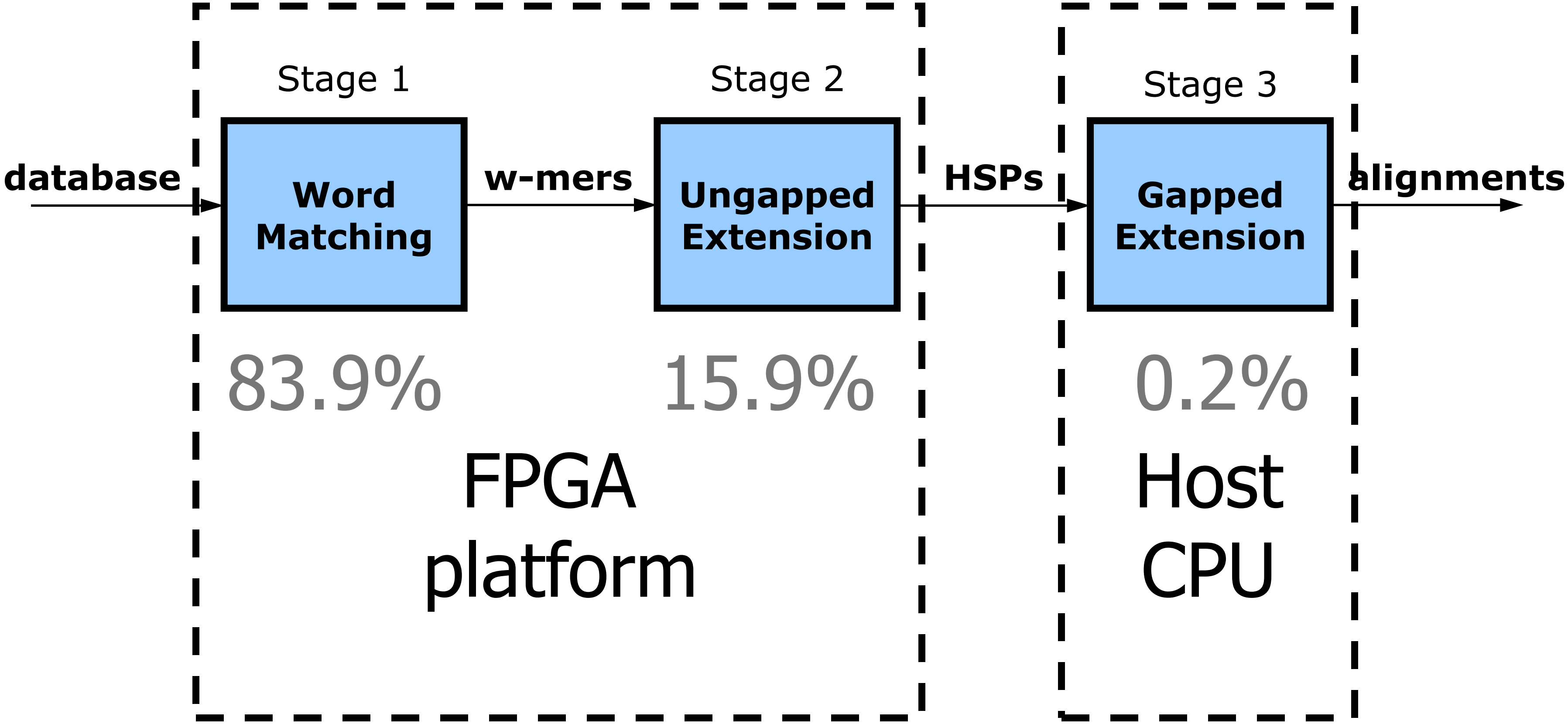
alignments

83.9%

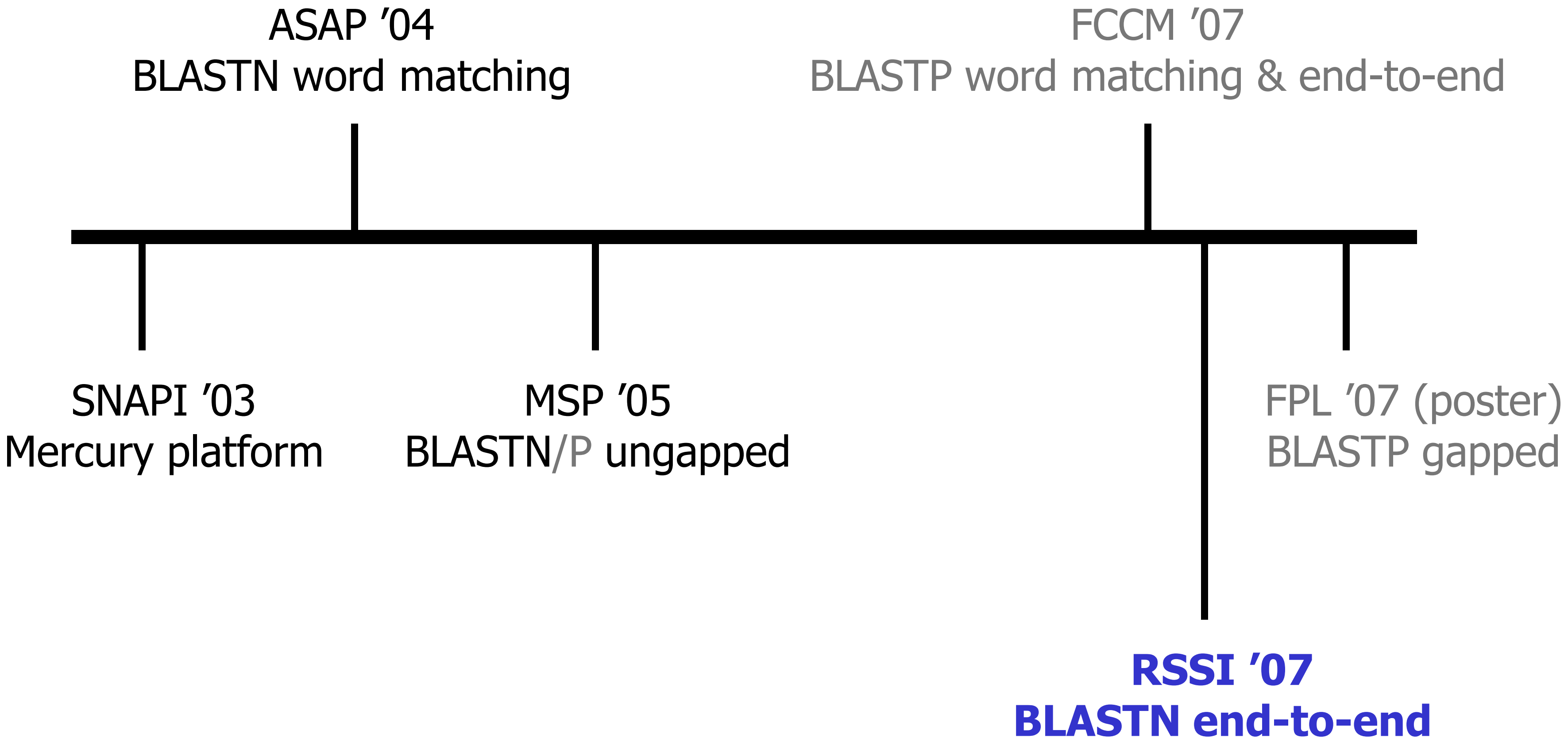
15.9%

0.2%

Hardware/Software Division



History of Mercury BLAST



SNAPI ’03
Mercury platform

ASAP ’04
BLASTN word matching

MSP ’05
BLASTN/P ungapped

FCCM ’07
BLASTP word matching & end-to-end

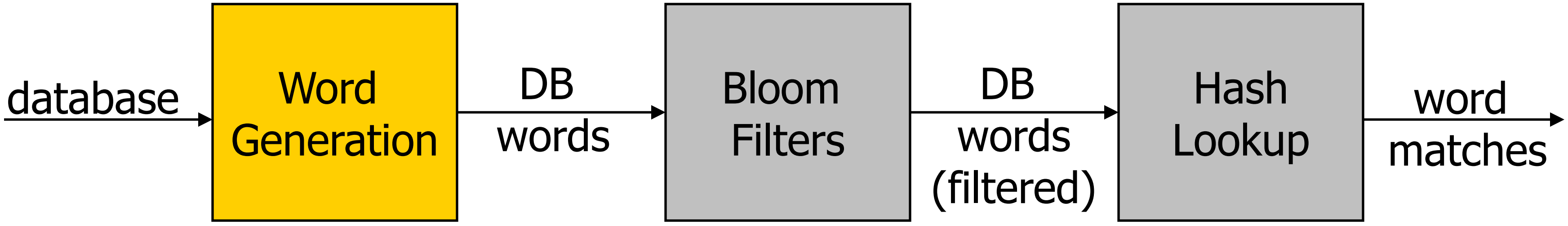
RSSI ’07
BLASTN end-to-end

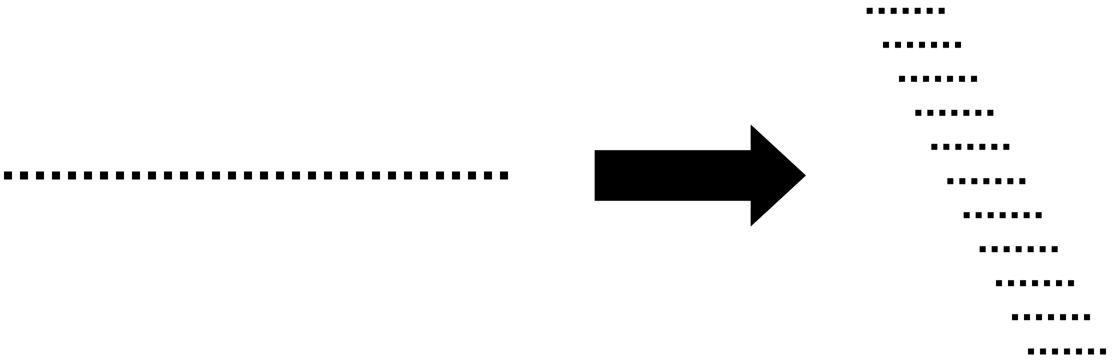
FPL ’07 (poster)
BLASTP gapped

Word Matching [K et al. '04]

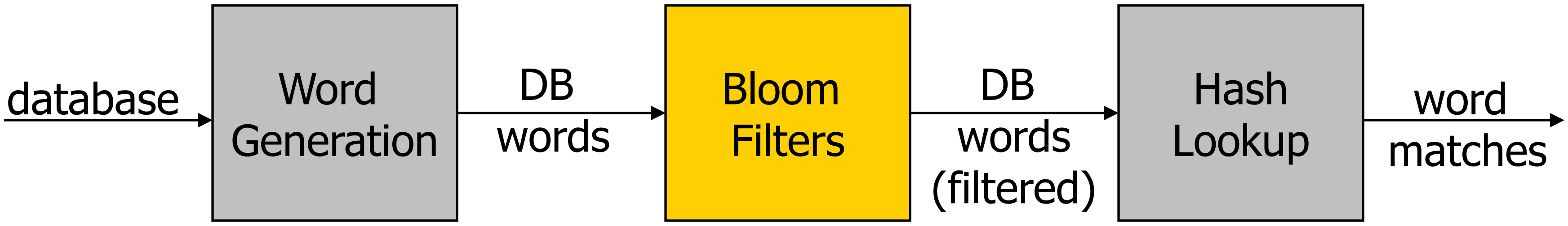
- Goal: find strings of length *w* in DB that also occur in query
- Basic approach: SRAM hash table built from query (limited bandwidth to FPGA!)
- Accelerant: Bloom filters on FPGA eliminate ~97% of lookups into hash table

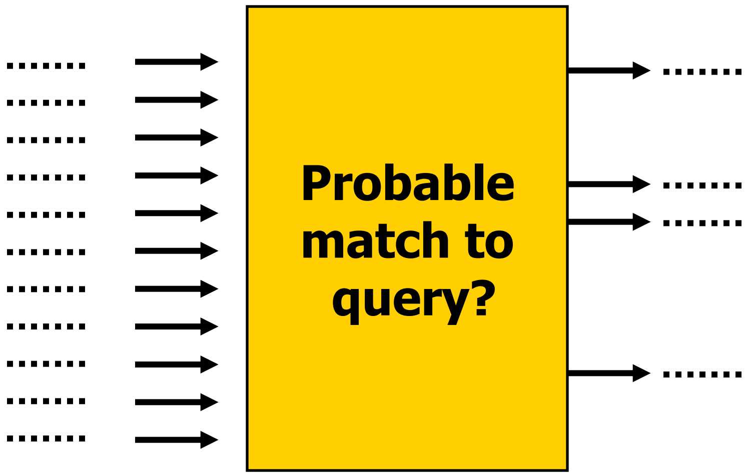
Stage 1 Execution



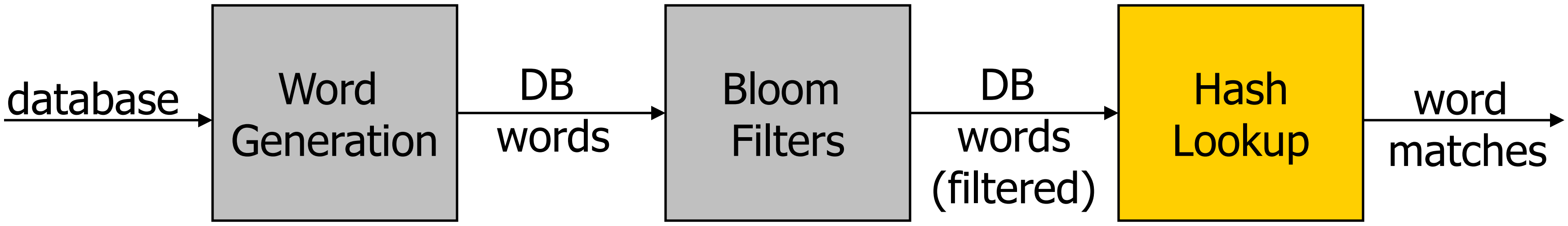


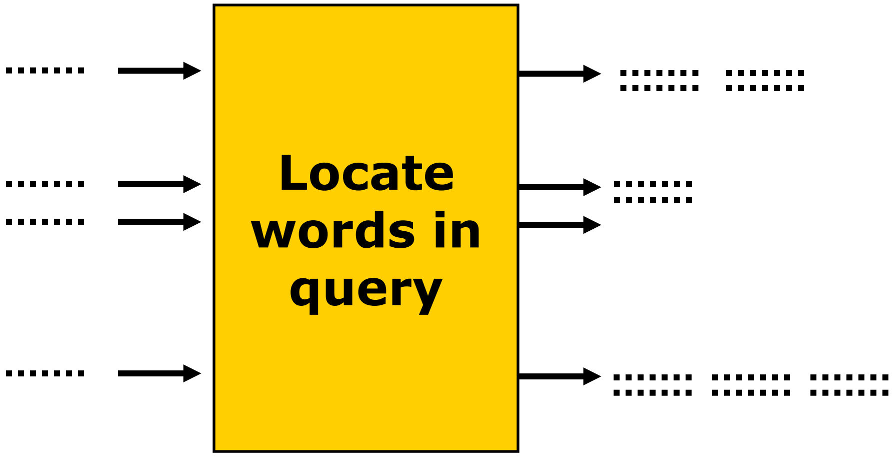
Stage 1 Execution





Stage 1 Execution



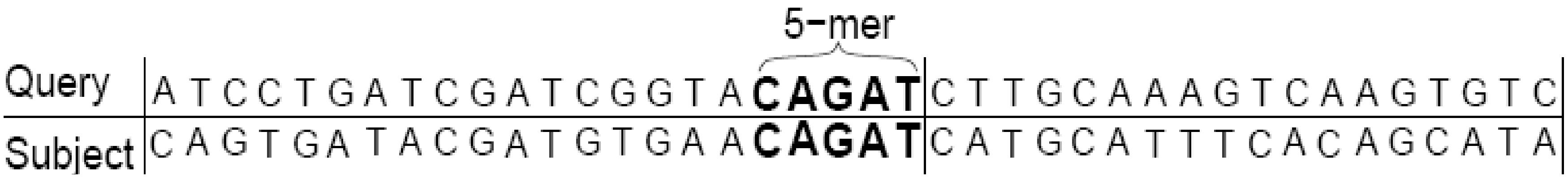


Ungapped Extension [L et al. ’05]

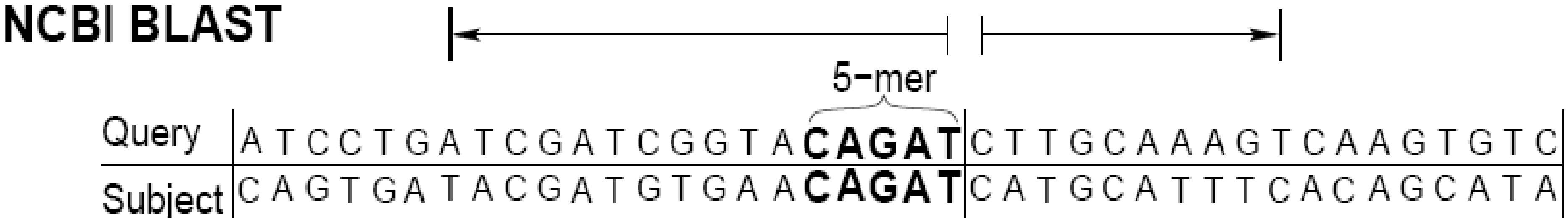
- Linear-time dynamic programming
- Systolic array design to pipeline DP
- DP limited to fixed-size window, unlike BLAST software

NCBI vs Mercury Ungapped Extension

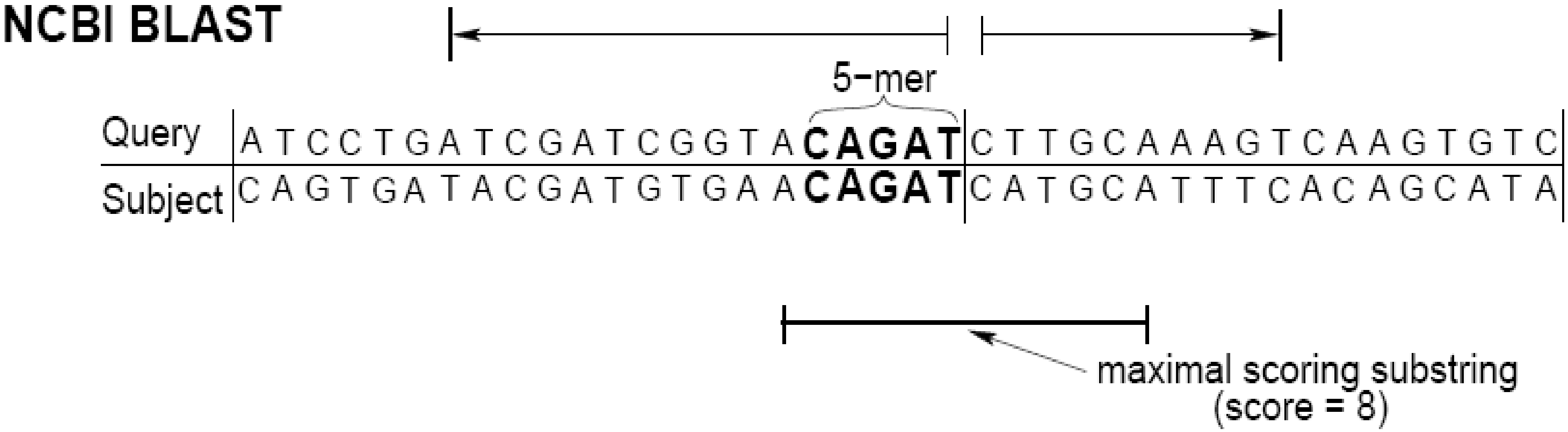
NCBI BLAST


5-mer
Query
ATCCTGATCGATCGGTACAGATCTTGCAAAGTCAAGTGTC
Subject
CAGTGATACGATGTGAACAGATCATGCATTTCACAGCATA

NCBI vs Mercury Ungapped Extension

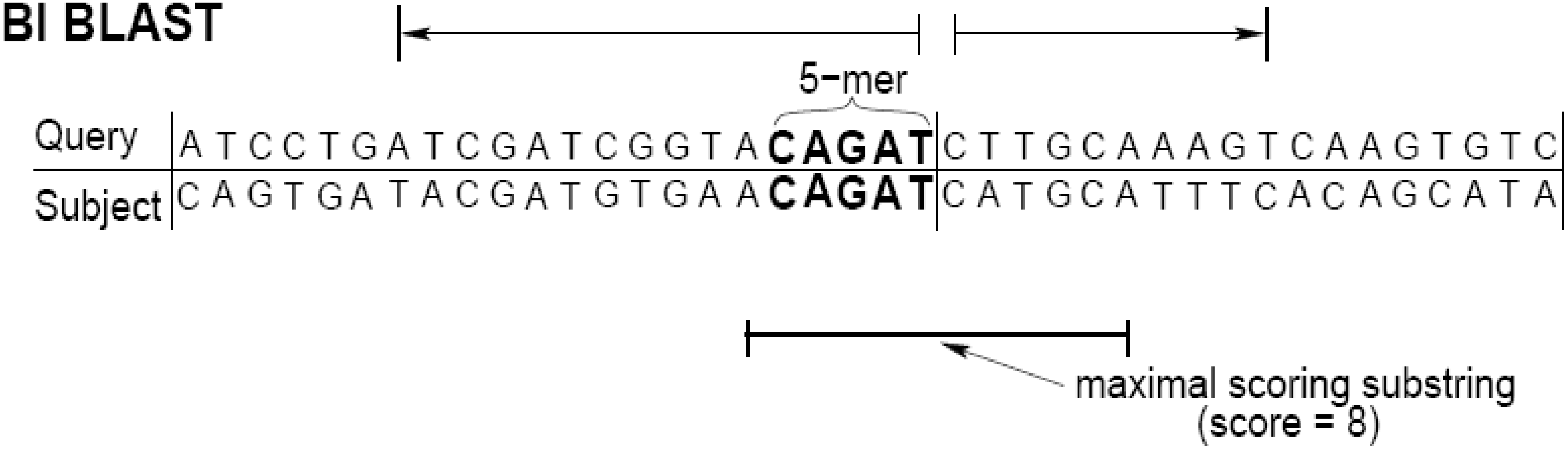

NCBI BLAST
5-mer
Query
ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC
Subject
CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

NCBI vs Mercury Ungapped Extension

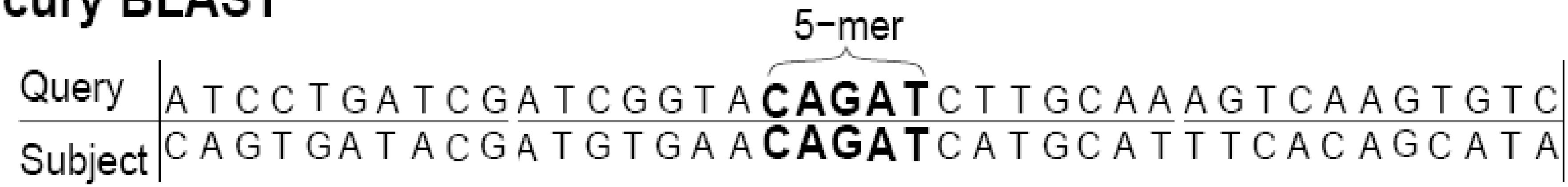

NCBI BLAST
5-mer
Query
A T C C T G A T C G A T C G G T A CAGAT C T T G C A A A G T C A A G T G T C
Subject
C A G T G A T A C G A T G T G A A CAGAT C A T G C A T T T C A C A G C A T A
maximal scoring substring
(score = 8)

NCBI vs Mercury Ungapped Extension

NCBI BLAST

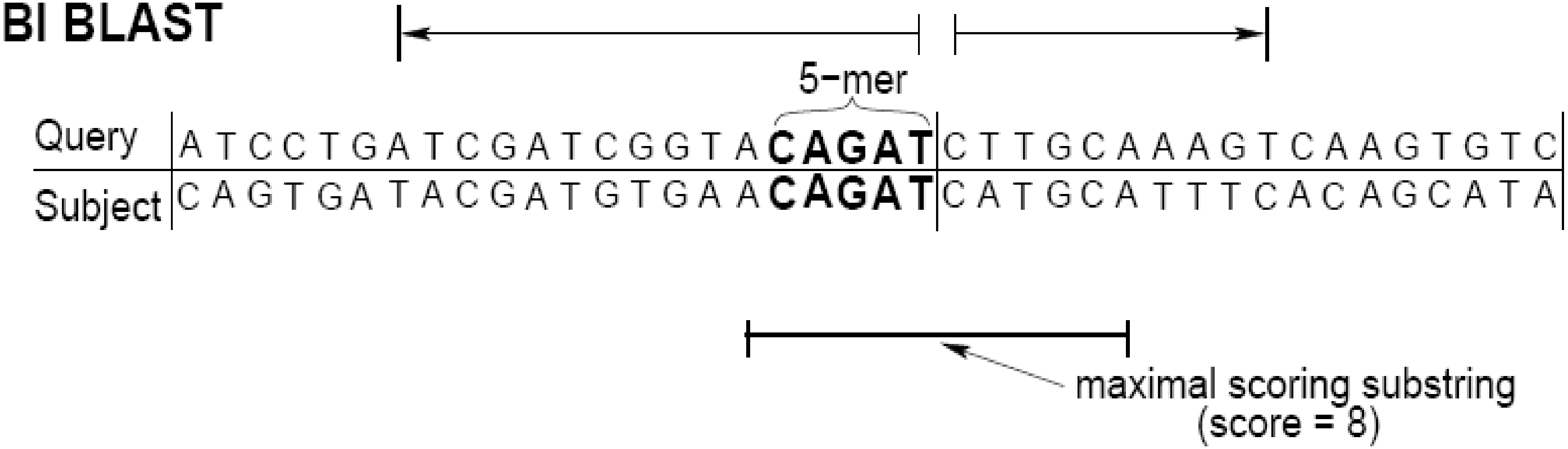

5-mer
Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC
Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA
maximal scoring substring
(score = 8)

Mercury BLAST


5-mer
Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC
Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

NCBI vs Mercury Ungapped Extension

NCBI BLAST



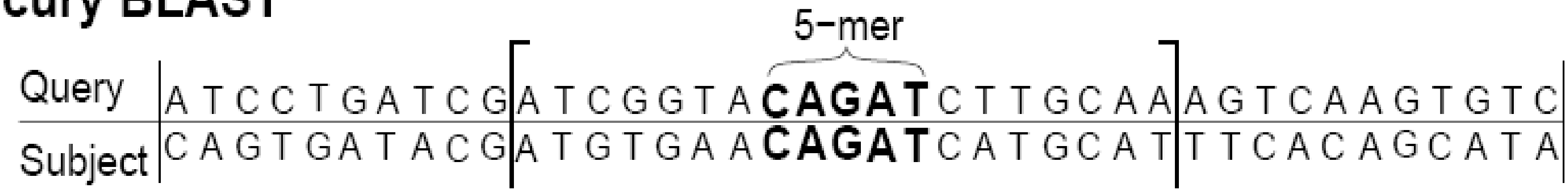
5−mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

maximal scoring substring
(score = 8)

Mercury BLAST

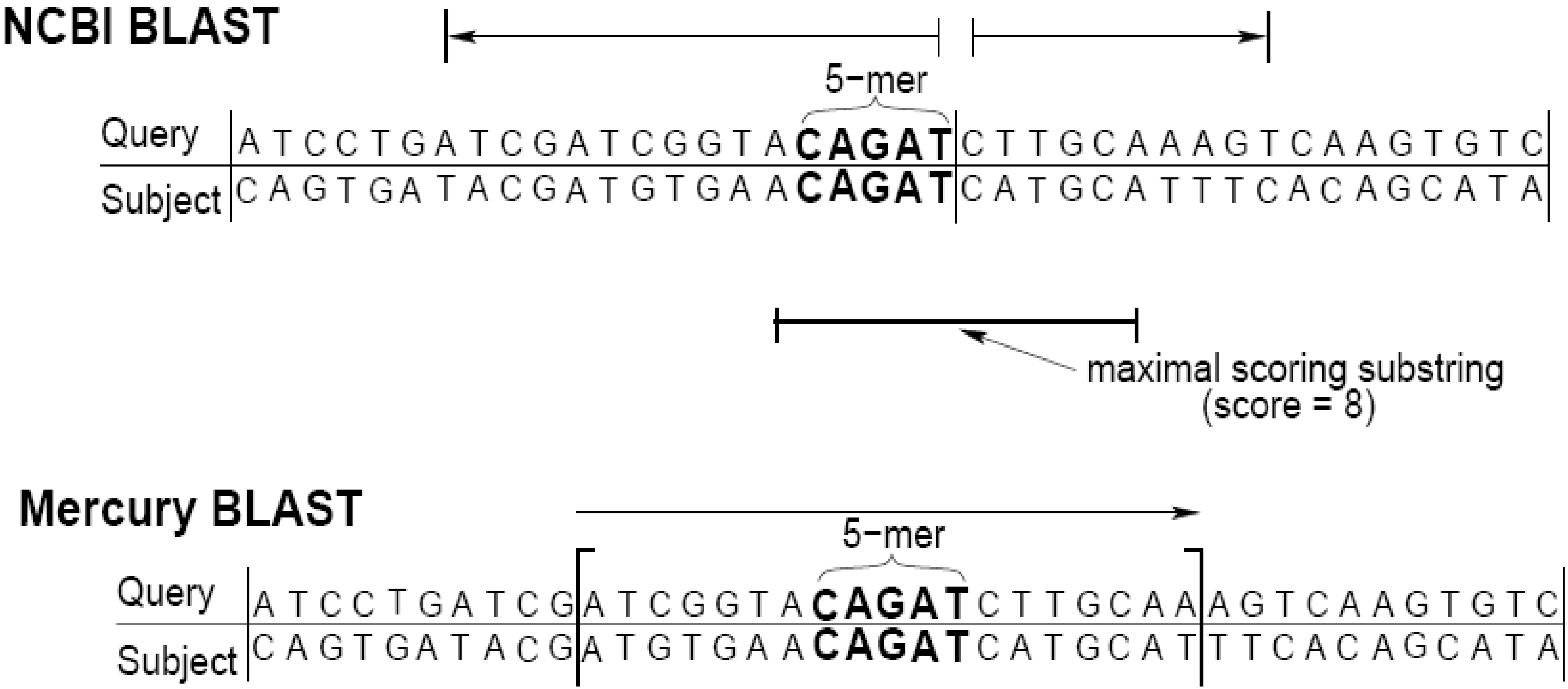


5−mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

NCBI vs Mercury Ungapped Extension



NCBI BLAST

5-mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

maximal scoring substring
(score = 8)

Mercury BLAST

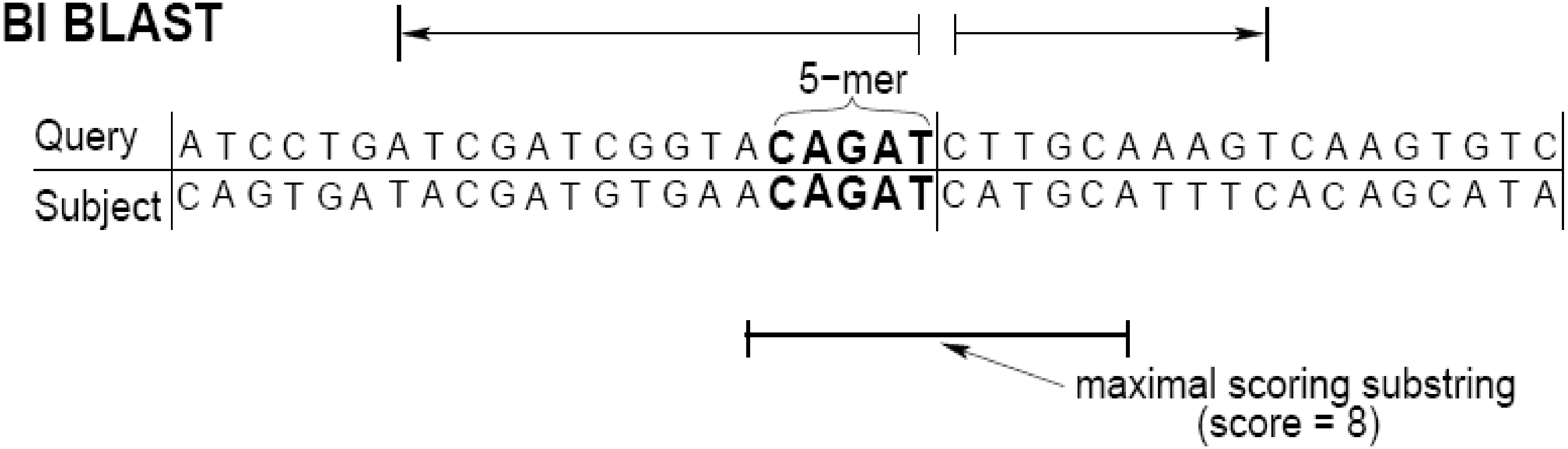
5-mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

NCBI vs Mercury Ungapped Extension

NCBI BLAST



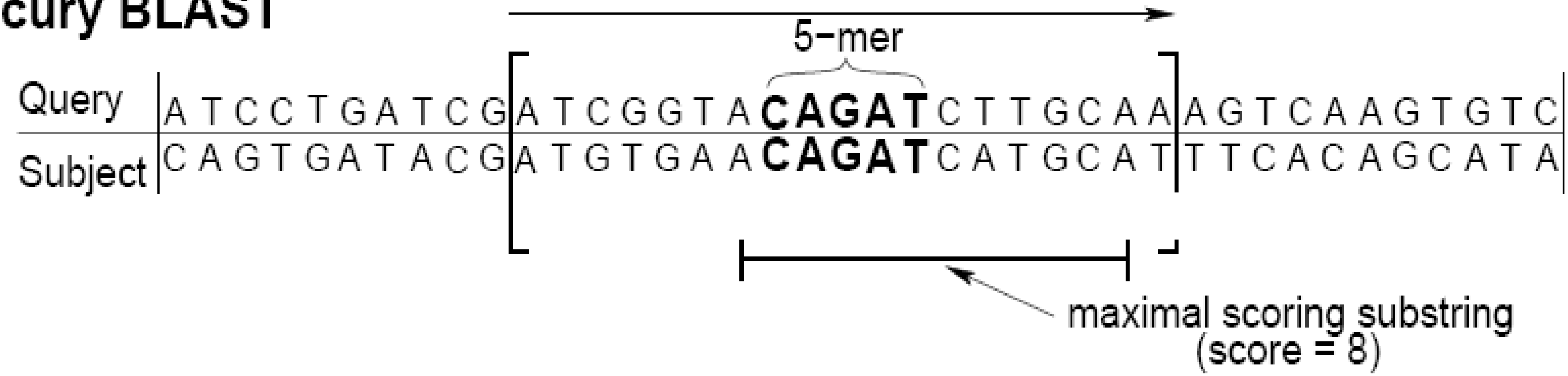
5-mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

maximal scoring substring (score = 8)

Mercury BLAST



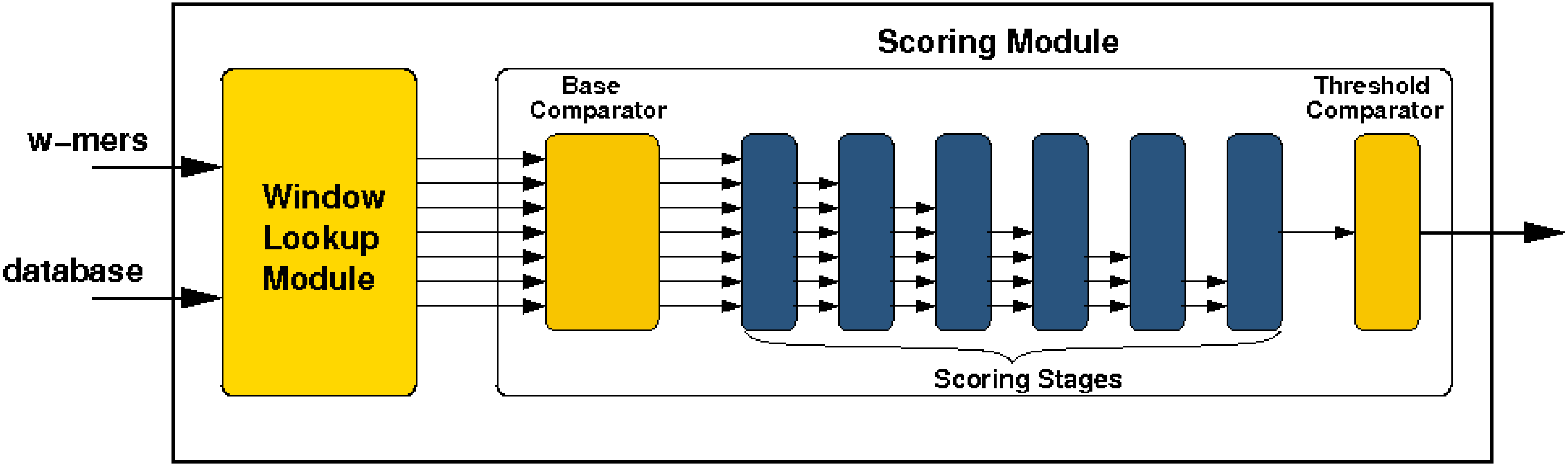
5-mer

Query ATCCTGATCGATCGGTA**CAGAT**CTTGCAAAGTCAAGTGTC

Subject CAGTGATACGATGTGAA**CAGAT**CATGCATTTCACAGCATA

maximal scoring substring (score = 8)

Stage 2 Architecture



extracts windows of query, DB to compare

scores of individual base match/mismatches

systolic array for DP

Is best ungapped alignment good enough to report?

Software Wrapper

- Front end, stage 3 use codebase of NCBI BLAST
- FPGA design replaces software stages 1 and 2
- Threads pipeline query prep, FPGA execution, and software stage 3 on different queries

Overview

- **Background** and Motivation
- **Methods**: Mercury BLASTN
- **Results**: end-to-end performance
- **Perspective**: opportunities for streaming computation on biosequences

Mercury BLASTN Implementation

- FPGA firmware
  - Functional modules coded in VHDL
  - running on Virtex II 6000-6 (AvNet devel board)
  - connected to host via PCI-X bus
  - comm. infrastructure by Exegy, Inc.
- Host system
  - dual 2.0 GHz AMD Opteron
  - (app uses < 10% of CPUs)
  - running Linux w/Exegy driver for FPGA
  - software based on NCBI BLASTN 2.2.10

Baseline for Comparison

- One core of Intel Pentium D 3.0 GHz
- ~one h/w generation newer than our FPGA board
- Running Linux
- NCBI BLASTN 2.2.15 (2.5x faster than 2.2.10!)

Experiment #1 –
mRNA vs mRNA (RefSeq v21)

- Q: 3975 human mRNAs (9 Mbase)
- DB: all other vertebrate mRNAs (586 Mbase)
- Med-low output stringency (E = 10^{-5})
- Why? Gene clustering, discovering variants in gene splicing across species

Results

Mercury BLASTN time	Speedup vs baseline	Total # alignments found	Overlap with baseline output
20 min	5.05x	6.2x10^{5}	98.64%

speed ~= 5 modern CPU cores

Experiment #2 – Genome vs Genome

- Q: Human chromosome 22 (21 Mbase)
- DB: mouse genome (1.5 Gbase)
- Med-low output stringency (E = 10^{-5})
- Why? Assigning orthology, detecting rearrangements

Results

Mercury BLASTN time	Speedup vs baseline	Total # alignments found	Overlap with baseline output
19 min	11.47x	9726	99.01%

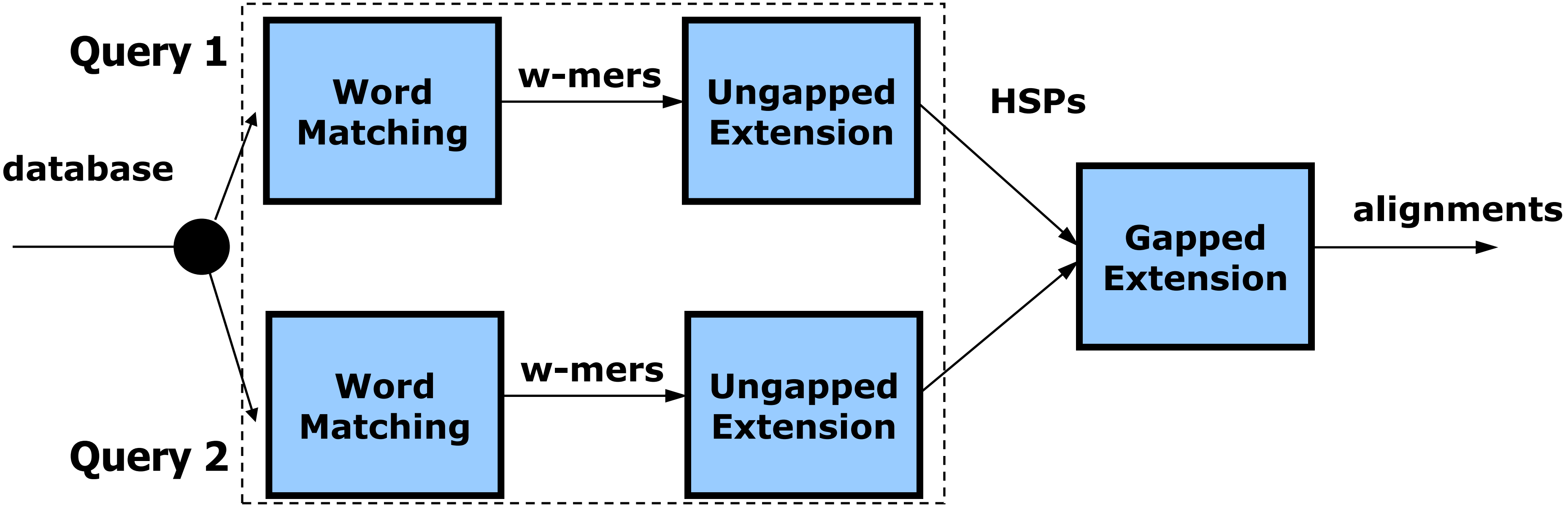
speed ~= 10 modern CPU cores

Where’s the Bottleneck?

- Each 17.5 kbase of query data requires one pass over whole database
- Query chunk size limited by stage 1 SRAM, Bloom filter blockRAM
- Each pass over DB saturates PCI-X link to card (> 700 Mbytes/sec)

How Will We Go Faster?

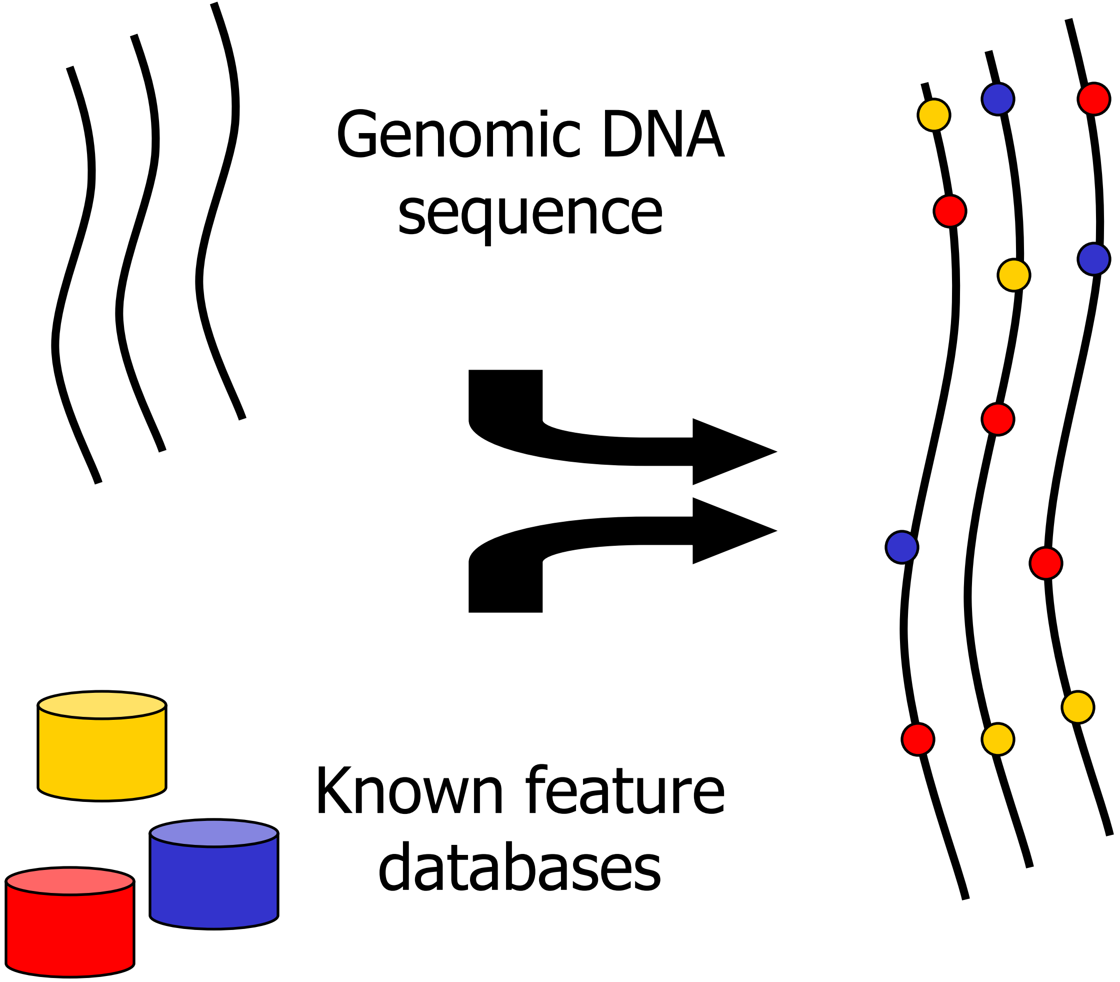
- New Exegy board: 2x Virtex 4 + SRAM
- Each core supports 4x larger query
- Hence, 8x more query per DB pass!



Overview

- **Background** and Motivation
- **Methods**: Mercury BLASTN
- **Results**: end-to-end performance
- **Perspective**: opportunities for streaming computation on biosequences

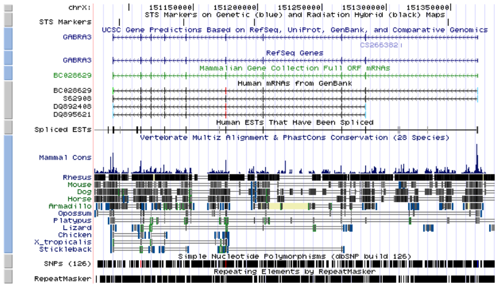
It’s All About Annotation



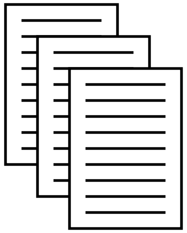
Genomic DNA sequence

Known feature databases

Annotated sequences


chrX: 151150000 151200000 151250000 151300000 151350000
STS Markers on Genetic (blue) and Radiation Hybrid (black) Maps
STS Markers
UCSC Gene Predictions Based on RefSeq, UniProt, GenBank, and Comparative Genomics
GABRA3
CS266382
RefSeq Genes
GABRA3
Mammalian Gene Collection Full ORF mRNAs
BC028629
Human mRNAs from GenBank
BC028629
S62908
DQ892408
DQ895621
Human ESTs That Have Been Spliced
Spliced ESTs
Vertebrate Multiz Alignment & PhastCons Conservation (28 Species)
Mammal Cons
Rhesus
Mouse
Dog
Horse
Armadillo
Opossum
Platypus
Lizard
Chicken
X_tropicalis
Stickleback
Simple Nucleotide Polymorphisms (dbSNP build 126)
SNPs (126)
Repeating Elements by RepeatMasker
RepeatMasker

data resources



insight

Generic Search Problem

- Given sequence(s) and DB of features…
- Label parts of sequence that are highly similar to some feature from DB
- Requires description of feature, measure of similarity

Generalized Features

- For BLAST, a feature is described by a single known sequence
- Can instead use a feature model that describes range of possible sequences
- (Typically a probabilistic model)

Typical Feature Models

Data	Model	Search Tool
DNA/protein aligned w/o gaps	PSSM	PSI-BLAST
DNA/protein aligned w/gaps	Profile HMM	HMMER
DNA/protein with evolutionary tree	phyloHMM	Phast (sort of)
RNA structure	SCFG	Infernal

Relevance of Mercury BLAST

- Many search apps look like BLAST
- Pipelined structure already present (PSI-BLAST) or could be designed (HMMER, Phast, Infernal)
- Mercury BLAST provides case study for how to accelerate these apps

Specific Challenges

- More complex measures of similarity (e.g. mutual information, phylogeny)
- Design filtering stages (like word matching) for newer DP-based tools
- Simplify FPGA development to serve limited application markets

Conclusions

- Order-of-magnitude BLASTN speedup, w/further 8x expected soon
- Answers 98.5%+ identical to software
- Design approach informs other high-performance biosequence search apps

Mercury BLAST Project

Faculty

- Jeremy Buhler
- Roger Chamberlain

Students

- Arpith Jacob
- Joe Lancaster
- Brandon Harris (graduated)
- Praveen Krishnamurthy (graduated)

Corporate Partners

- BECS Technology, Inc.
- Exegy, Inc.

Funding Agencies

- NIH NHGRI
- NSF BIO
- NSF CISE

Thank You!