L’art de rechercher et de comparer des séquences biologiques

Luca Freschi

Club Bioinfo IBIS

The fastest tool to search sequences

Who will set the World Record?

The contenders.

🟢 Blast	
🔴 Usearch	
🔵 Last	
🟣 Diamond	

● Blast

Age: 26 (born: 1990) — last release: 2015

Father(s)/Mother(s): Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ

Team: NCBI

Home: http://blast.ncbi.nlm.nih.gov

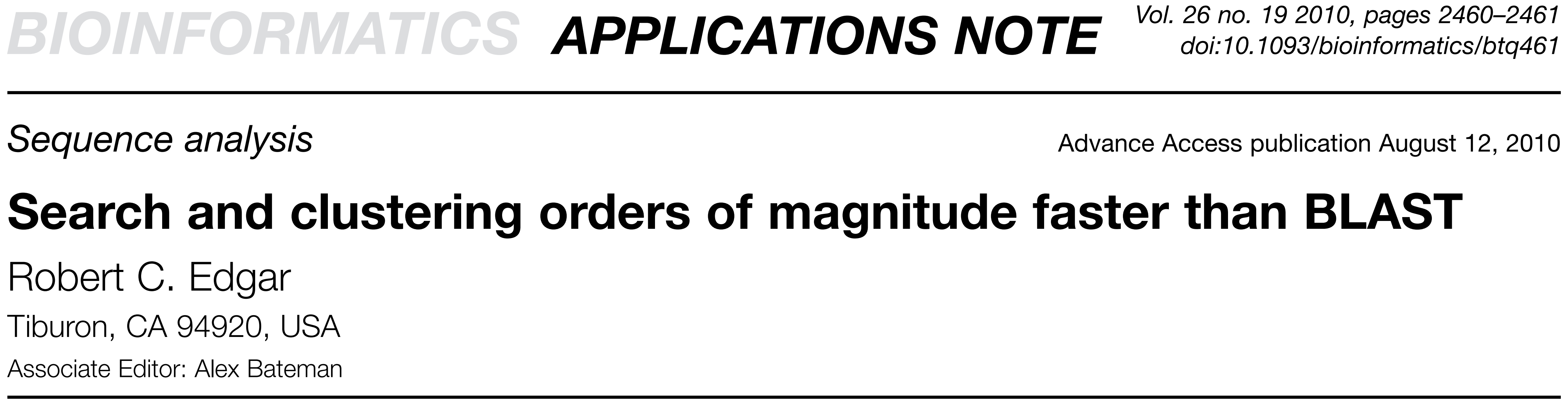
● Usearch

Age: 6 (born: 2010) — last release: 2015

Father(s)/Mother(s): Robert C. Edgar

Team: himself — independent researcher

Home: http://drive5.com/usearch/



BIOINFORMATICS APPLICATIONS NOTE

Vol. 26 no. 19 2010, pages 2460–2461
doi:10.1093/bioinformatics/btq461

Sequence analysis

Advance Access publication August 12, 2010

Search and clustering orders of magnitude faster than BLAST

Robert C. Edgar
Tiburon, CA 94920, USA

Associate Editor: Alex Bateman

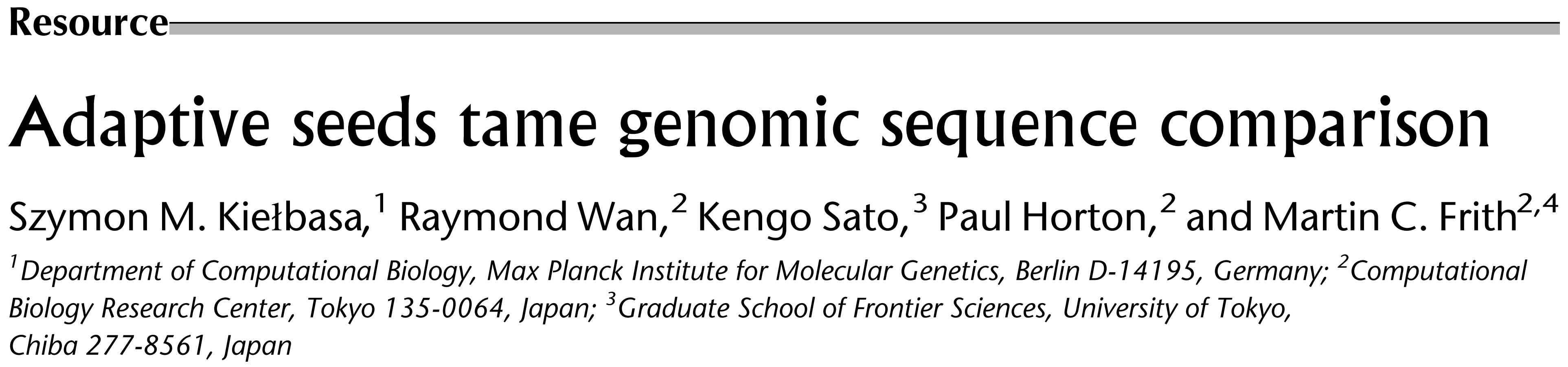
● Last

Age: 5 (born: 2011) — last release: 2016

Father(s)/Mother(s): Kiełbasa SM$_1$, Wan R, Sato K, Horton P, Frith MC

Team: University of Tokyo

Home: http://last.cbrc.jp/



Resource

Adaptive seeds tame genomic sequence comparison

Szymon M. Kiełbasa,[1] Raymond Wan,[2] Kengo Sato,[3] Paul Horton,[2] and Martin C. Frith[2,4]

[1]Department of Computational Biology, Max Planck Institute for Molecular Genetics, Berlin D-14195, Germany; [2]Computational Biology Research Center, Tokyo 135-0064, Japan; [3]Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-8561, Japan

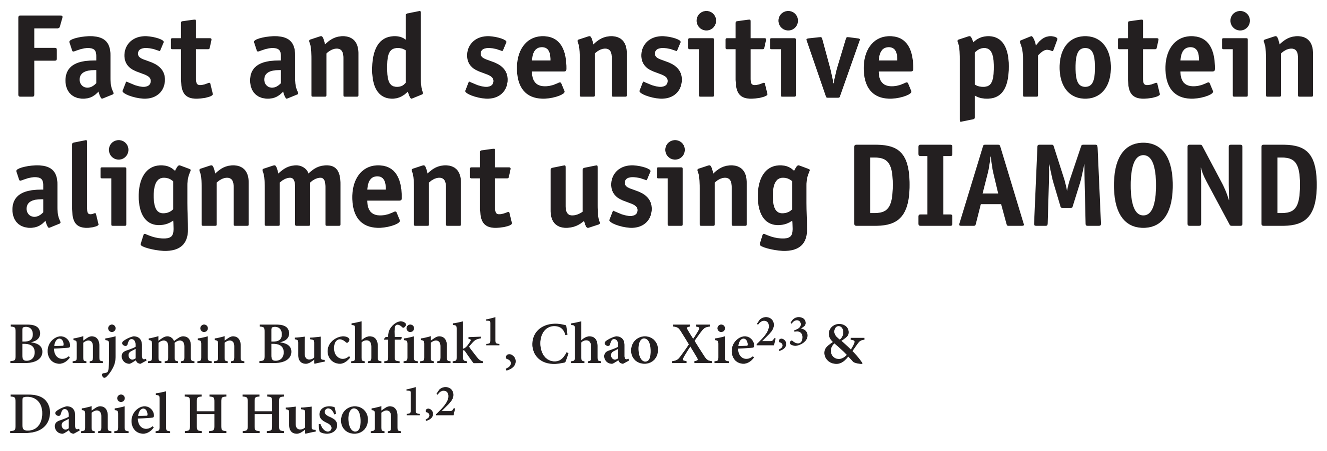
Diamond

Age: 1 (born: 2015) — last release: 2016

Father(s)/Mother(s): Buchfink B, Xie C. & Huson Daniel H

Team: University of Tübingen

Home: https://github.com/bbuchfink/diamond

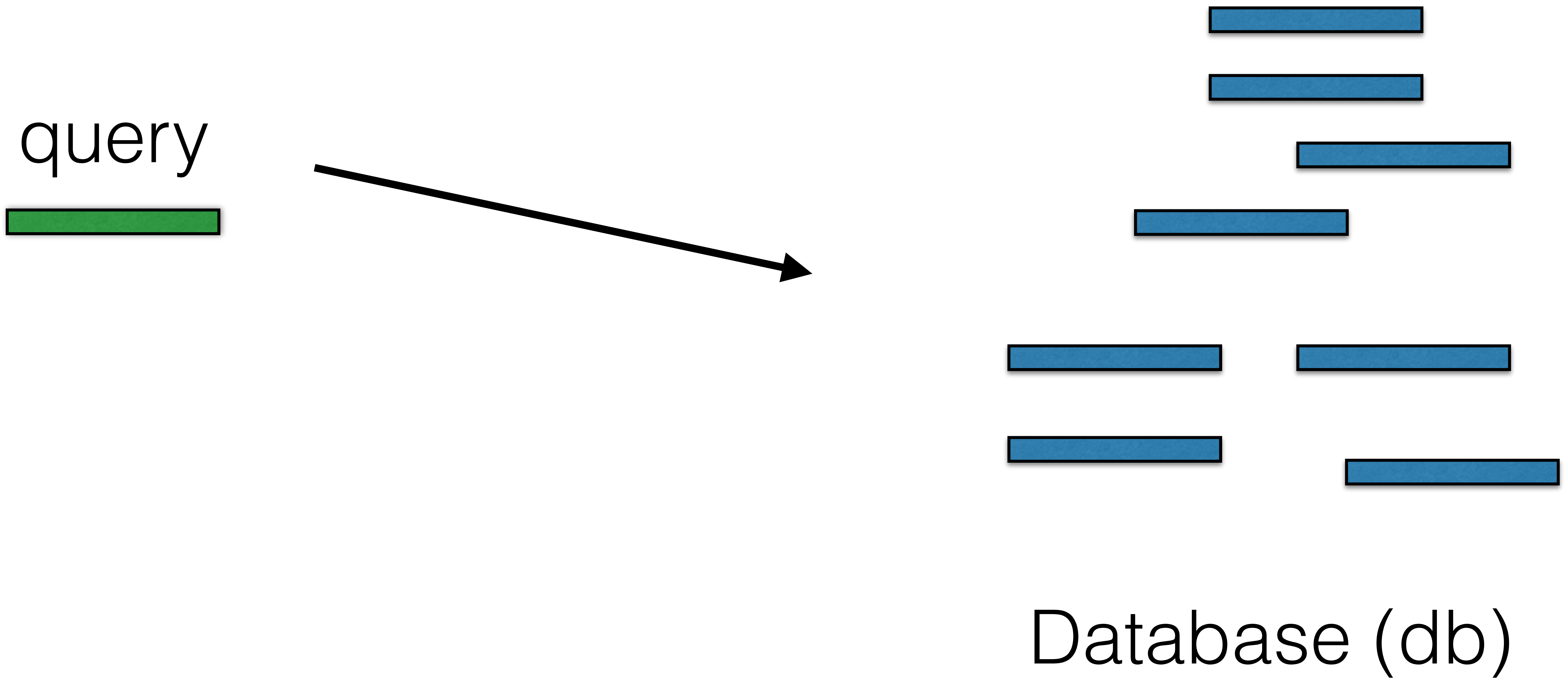


Fast and sensitive protein alignment using DIAMOND

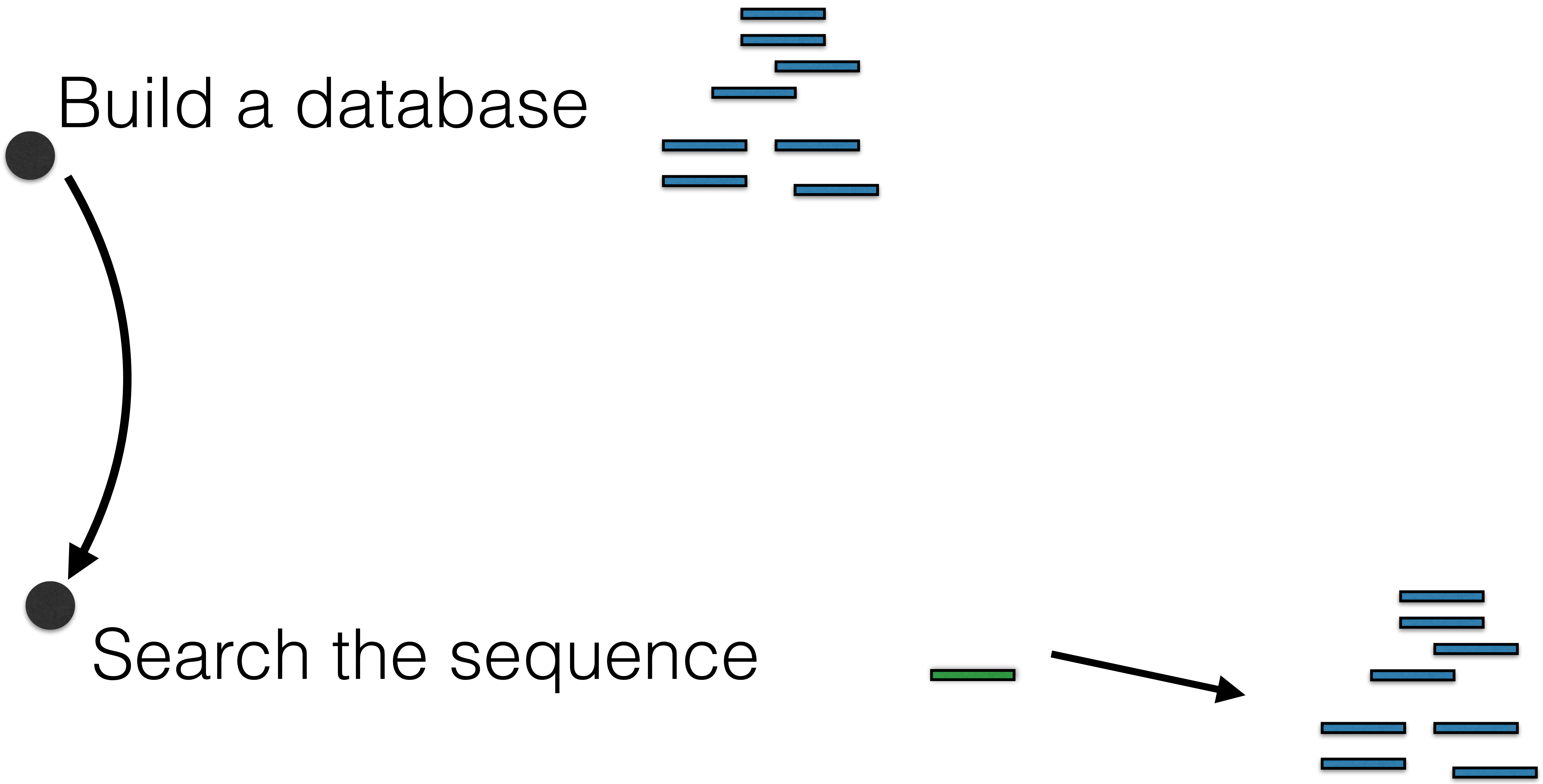
Benjamin Buchfink[1], Chao Xie[2,3] &
Daniel H Huson[1,2]

Published in *Nature Methods* **12**, 59–60 (2015) | doi:10.1038/nmeth.3176

General steps required to perform a sequence search



General steps required to perform a sequence search



Build a database

Search the sequence

The blast tabular output

```
qseqid          sseqid                 pident  length mismatch gapopen qstart  qend    sstart   send    evalue  bitscore

ipcd241_seq1    000006765.1_seq5525     99.58   237     1       0       1       237     182     418     6e-173    484
ipcd241_seq1    000006765.1_seq2497     88.09   235     28      0       1       235     182     416     4e-145    414
ipcd241_seq1    000006765.1_seq4697     78.30   235     51      0       1       235     181     415     8e-137    392
ipcd241_seq1    000006765.1_seq2447     46.88   32      15      2       41      71      256     286     1.7     26.9
ipcd241_seq1    000006765.1_seq4359     31.11   45      31      0       23      67      278     322     2.8     26.6
ipcd241_seq1    000006765.1_seq1583     33.75   80      38      3       149     228     148     212     3.1     26.2
ipcd241_seq1    000006765.1_seq4908     50.00   28      12      1       40      67      282     307     6.7     25.4
ipcd241_seq1    000006765.1_seq832      26.56   64      41      2       9       69      190     250     8.6     25.0
```

BLAST

● Build a database

```
makeblastdb -in <faa_file> -dbtype prot
```

● Search the sequence

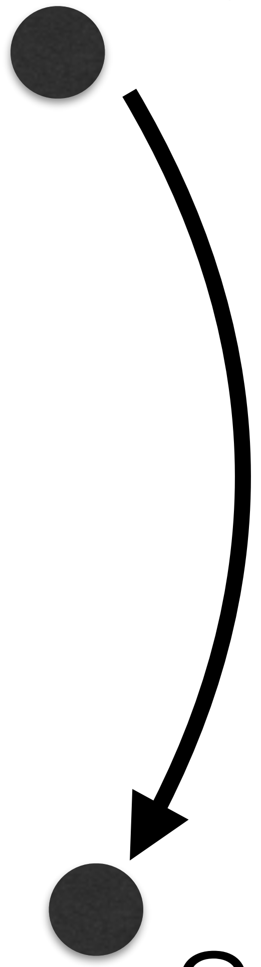
```
blastp -query <fasta_query> -db <db_name> -out <out_file> -outfmt 6
```

The BLAST output

```
qseqid          sseqid                 pident  length mismatch gapopen qstart  qend    sstart   send    evalue  bitscore

ipcd241_seq1    000006765.1_seq5525     99.58   237     1       0       1       237     182     418     6e-173    484
ipcd241_seq1    000006765.1_seq2497     88.09   235     28      0       1       235     182     416     4e-145    414
ipcd241_seq1    000006765.1_seq4697     78.30   235     51      0       1       235     181     415     8e-137    392
ipcd241_seq1    000006765.1_seq2447     46.88   32      15      2       41      71      256     286     1.7     26.9
ipcd241_seq1    000006765.1_seq4359     31.11   45      31      0       23      67      278     322     2.8     26.6
ipcd241_seq1    000006765.1_seq1583     33.75   80      38      3       149     228     148     212     3.1     26.2
ipcd241_seq1    000006765.1_seq4908     50.00   28      12      1       40      67      282     307     6.7     25.4
ipcd241_seq1    000006765.1_seq832      26.56   64      41      2       9       69      190     250     8.6     25.0
```

Usearch



Build a database

```
usearch8 -makeudb_usearch <faa_file> -output <db_name>
```

Search the sequence

```
usearch8 -usearch_local <faa_file>  -db <db_name> -id <identity>  -blast6out <file_out>
```

The Usearch output

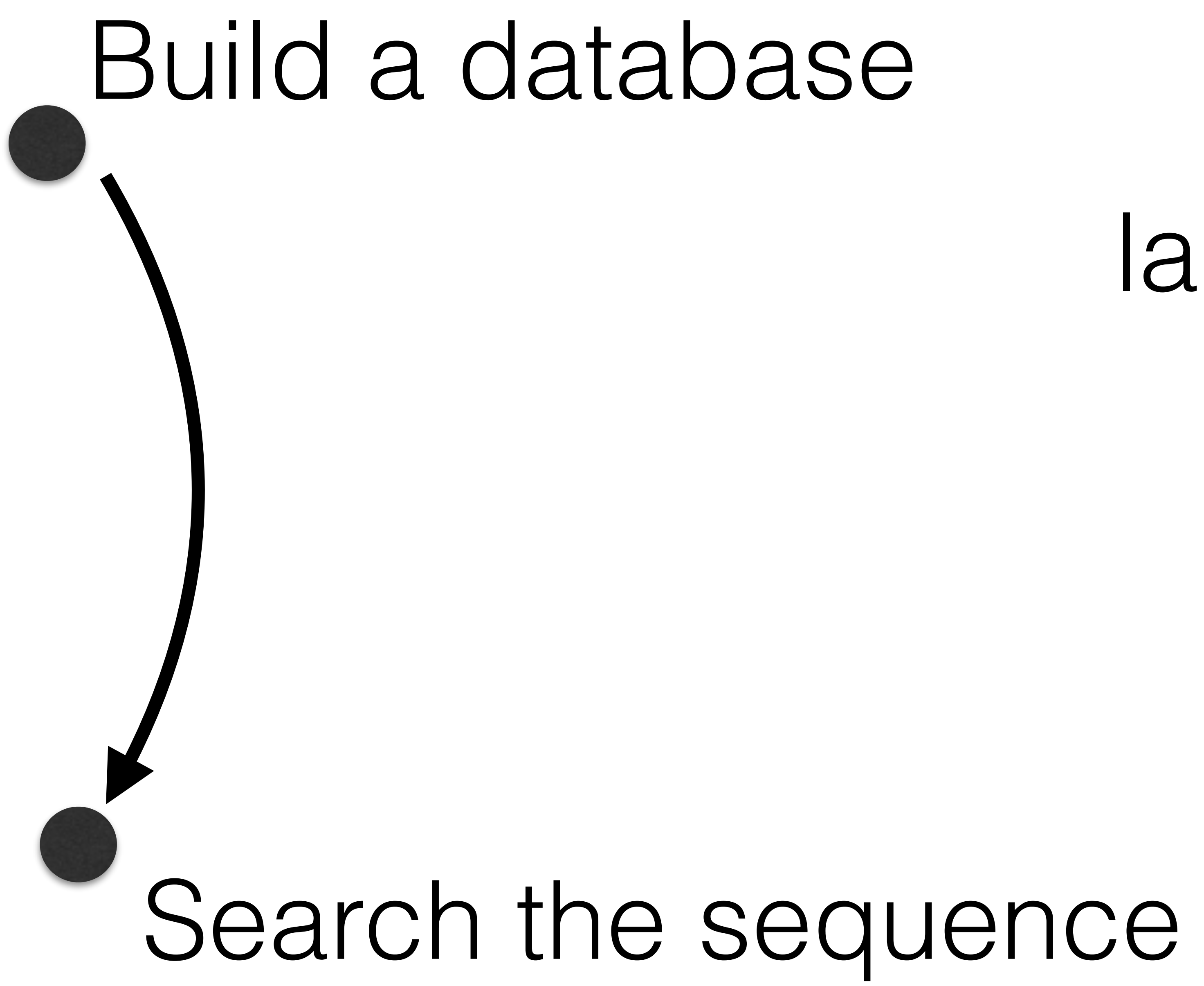
```
qseqid          sseqid                     pident  length mismatch gapopen qstart  qend    sstart  send     evalue          bitscore
 ipcd241_seq1    000006765.1_seq5525         99.6     236      1        0        1       236     182     417     3.4e-133          468.8
```

Let’s increase the accepted sequences

```
usearch8 -usearch_local <faa_file>  -db <id> -id <identity> -maxaccepts <num_accpt_seq>  -blast6out <file_out>
```

```
qseqid          sseqid                 pident  length mismatch gapopen qstart  qend    sstart   send     evalue       bitscore
ipcd241_seq1    000006765.1_seq5525     99.6    236     1        0       1       236    182     417     3.4e-133       468.8
ipcd241_seq1    000006765.1_seq2497     88.1    235     28       0       1       235    182     416     5.8e-117       414.8
ipcd241_seq1    000006765.1_seq4697     78.3    235     51       0       1       235    181     415     1.6e-106       380.2
```

Last



Build a database

```
lastdb -p <db_name> <faa_file>
```

Search the sequence

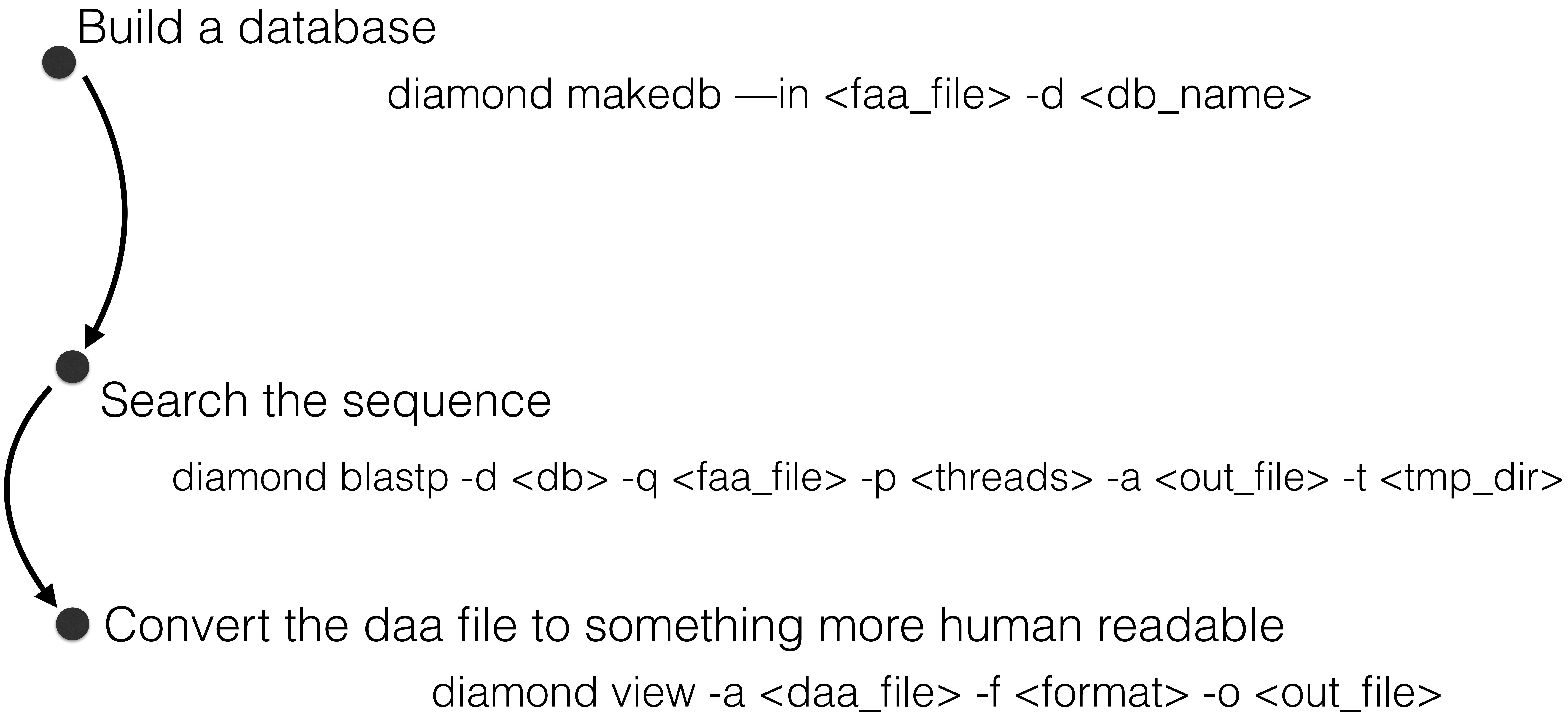
```
lastal -f <format> <db_name> <faa_file>><out_file>
```

The Last output

```
qseqid          sseqid                    pident  length mismatch gapopen qstart  qend    sstart   send    evalue       bitscore

ipcd241_seq1   000006765.1_seq5525       99.58    237       1         0       1       237      182     418    4.5e-174         536
ipcd241_seq1   000006765.1_seq2497       88.09    235       28        0       1       235      182     416    2.3e-152         473
ipcd241_seq1   000006765.1_seq4697       78.30    235       51        0       1       235      181     415    2.7e-138         434
```

● Diamond



Build a database

```
diamond makedb —in <faa_file> -d <db_name>
```

Search the sequence

```
diamond blastp -d <db> -q <faa_file> -p <threads> -a <out_file> -t <tmp_dir>
```

Convert the daa file to something more human readable

```
diamond view -a <daa_file> -f <format> -o <out_file>
```

The Diamond output

```
qseqid          sseqid                    pident  length  mismatch  gapopen  qstart  qend  sstart  send  evalue     bitscore
ipcd241_seq1    000006765.1_seq5525       99.6    237     1         0        1       237   182     418   2.6e-133   469.2
ipcd241_seq1    000006765.1_seq2497       88.1    235     28        0        1       235   182     416   5.8e-117   414.8
ipcd241_seq1    000006765.1_seq4697       78.3    235     51        0        1       235   181     415   1.6e-106   380.2
```

On your marks, set, go!

Contenders:

- 🟢 Blast (R)
- 🔴 Usearch
- 🔵 Last
- 🟣 Diamond

Challenge:

query (18)

db (1)

On your marks, set, go!

Contenders:	Indexing:	Searches:
🟢 Blast (R)	00:00:10	03:25:34
🔴 Usearch		
🔵 Last		
🟣 Diamond		

On your marks, set, go!

Contenders:	Indexing:	Searches:
🟢 Blast (R)	00:00:10	03:25:34
🔴 Usearch		
🔵 Last		
🟣 Diamond	00:00:06	00:08:54

On your marks, set, go!

Contenders:	Indexing:	Searches:
🟢 Blast (R)	00:00:10	03:25:34
🔴 Usearch	00:00:27	00:00:48
🔵 Last		
🟣 Diamond	00:00:06	00:08:54

On your marks, set, go!

Contenders:	Indexing:	Searches:
🟢 Blast (R)	00:00:10	03:25:34
🔴 Usearch	00:00:27	00:00:48
🔵 Last	00:00:06	
🟣 Diamond	00:00:06	00:08:54

On your marks, set, go!

Contenders:	Indexing:	Searches:
● Blast (R)	00:00:10	03:25:34
● Usearch	00:00:27	00:00:48 ✓
● Last	00:00:06	00:01:41
● Diamond	00:00:06	00:08:54

Lets push last at his limit!

-k option: By default lastal looks for initial matches starting at every position in the query sequence(s), but -k2 makes it check every 2nd position, -k3 every 3rd position, etc. Compared to the other sparsity options, this **increases speed** the most while **reducing sensitivity** the least.

Lets push last at his limit!

Contenders:	Indexing:	Searches:
● Blast (R)	00:00:10	03:25:34
● Usearch	00:00:27	00:00:48
● Last	00:00:06	00:01:41
● Diamond	00:00:06	00:08:54
● Last -k 10	00:00:07	00:00:34 ✓

Estimating the overlap between the searches

		Overlap with BLAST
🟢 blast:	120801 links	100.0%
🔴 usearch:	111349 links	91.1%
🔵 last:	112941 links	92.4%
🟣 diamond:	112435 links	91.6%
🔵 last_k-10:	112092 links	91.6%