# LISA

**Install**

do CXX=icpc at make
add -I bwa to the makefile (no need for the latest commit)
add -std=c++1y to line 21 in scripts/build-rmi.linear_spline.linear.sh
install rust
use absolute path when building the index

**exact-search-lisa.o**

./exact-search-lisa.o /nfshomes/yhxu/scratch/858D/data/fasta/test.fasta /nfshomes/yhxu/scratch/858D/data/query/test.query 3 3 1

./exact-search-lisa.o /nfshomes/yhxu/scratch/858D/858D-project/data/fasta/Chlamydia.fasta /nfshomes/yhxu/scratch/858D/858D-project/data/query/Chlamydia.32.query 4 256 1

benchmarks/bench-fixed-len-e2e-match-lisa.cpp

- [62] main
- [81] read reference sequence. Printed out and it is in ACGT form and is correct
- [86] read in queries. Printed out and it is in ACGT form and concatenated by ; and is correct
- [105] add $ to the reference seq
- [124] create rmi
- src/ipbwt_rmi.h
    - [226] constructor
    - [234 - 332] define a function to build ipbwt from scratch
    - [339] check if ipbwt already exists. If it is, load it at [350]. If not, build it from scratch at [367]. **The code will fail at this place if it doesn't exist.**
        - [234] call build_ipbwt
        - **[240] build __sa**
        - **[265] free __sa**
        - **should save __sa here**
        - [322] sanity check, it will fail at the last entry
    - [375] check if rmi parameters exists. If it is, load it at [384]. If not, evoke scripts to train it [405 - 438]
- [124] With pre-built index, rmi should return correctly
- [162] create threads
- [196] go through all queries with while loop.

        - **The code will end up with infinite loop if query length is not multiple of k**
        - **Fixed by making interation condition to be num_iter > 0**
        - The code give wrong result if query length is not multiple of k
        - If query lengths are different, the code give wrong answers for all queries except the longest ones.
- [233] program failed because of float point exception at the #pragma
    - **fixed, it's because parallel_batch_size = 0 when there are few queries**
- [251] call rmi.backward_extend_chunk_batched(&str_enc[i], qs_sz, &intv_all[i*2]);
- src/ipbwt_rmi.h
    - [956] backward_extend_chunk_batched
    - [982] call process_query_one_step
    - [850] process_query_one_step
    - [923] call last_mile_vectorized_search_final_step(meta.ipb_x[0], meta.first[0], meta.m[0]);
    - [824] last_mile_vectorized_search_final_step
    - [827] call _mm512_loadu_si512, causes illegal instruction error

**build-index-forward-only-lisa.o**

./build-index-forward-only-lisa.o /nfshomes/yhxu/scratch/858D/data/fasta/test.fasta 3 3
./build-index-forward-only-lisa.o /nfshomes/yhxu/scratch/858D/data/fasta/Ecoli.fasta 3 3

Mostly same as querying. Main code is in

- src/ipbwt_rmi.h [226-438]

**IPBWT size:**

src/ipbwt_rmi.h

- [60, 63, 65] Hard-code NUM_POS_BITS = 38, NUM_CHUNK_BITS = 42, NUM_IPBWT_BYTES = (NUM_IPBWT_BITS + 7 ) / 8 = 10
- [289] allocate n * NUM_IPBWT_BYTES for IPBWT
- [323 - 345] computing IPBWT
- [334] encode kmer
- [341 - 344] put kmer and positions into IPBWT

# Pufferfish

**pufferfish index**

pufferfish index -r <ref_file>... -o <output_dir> [--expectTranscriptome] [--headerSep <sep_strs>] [--keepFixedFasta] [--keepDuplicates] [-d <decoy_list>] [-n] [-f <filt_size>] [--tmpdir <twopaco_tmp_dir>] [-k <kmer_length>] [-p <threads>]

./src/pufferfish index -r /Users/henryxu/Desktop/Sp2022/858D/project/data/fasta/test.fasta -o /Users/henryxu/Desktop/Sp2022/858D/project/data/pufferfish-index/test/ -k 3

Pufferfish.cpp

- [46] main
- [313] call pufferfishIndex(indexOpt)
- PufferfishIndexer.cpp
  - [355] `pufferfishIndex(pufferfish::IndexOptions& indexOpts)`
  - [457] print `ntHll estimated {} distinct k-mers, setting filter size to`
  - [499] call `buildGraphMain(args)` to use TwoPaCo to build the compacted dbg
  - [523] call `dumpGraphMain(args)` to use TwoPaCo to serialize the compacted dbg
  - [539,540] reading GFA file created by TwoPaco and then use it them to create contig table `pufferfish::BinaryGFAReader pf(outdir.c_str(), k - 1, buildEqCls, buildEdgeVec, jointLog)`
  - PufferfishBinaryGFAReader.cpp
    - [84] constructor
    - [96] read in the contig sequence created by TwoPaCo. Here we can print out the contig
    - [101] read in the bv created by TwoPaCo. Here we can print out the bv
    -

**pufferfish kquery**

pufferfish kquery -i <index> -q <ref>... [-p <threads>] [-v]

./src/pufferfish kquery -i
/Users/henryxu/Desktop/Sp2022/858D/project/data/pufferfish-index/test/ -q
/Users/henryxu/Desktop/Sp2022/858D/project/data/query/test.query

Pufferfish.cpp

- [46] main
- [313] call pufferfishKmerQuery(kmerQueryOpt)
- PufferfishKmerQuery.cpp
    - [112] pufferfishKmerQuery
    - [150] load pufferfishIndex
    - pufferfishIndex.cpp
        - [115] load contiq sequence “seq_”
    - [156] call doPufferfishKmerQuery(pi, parser, iomut)
    - [25] doPufferfishKmerQuery
    - [65] call auto phits = pi.getRefPos(km, qc);
    - PufferfishIndex.cpp
        - [158] PufferfishIndex::getRefPos(CanonicalKmer& mer, pufferfish::util::QueryCache& qc)
        - [163] look up MPH table
        - [164] check this kmer exists
        - [165] look up pos vector to get the position in contig sequence ‘seq_’
        - This can be replaced by querying LISA FM-Index