**SUPPLEMENTAL INFORMATION**

Supplemental Information includes four tables, three figures, one dataset and a method.

Supporting Information Legends

S1: Sequence data source

S2: CG and AG (purine) content percentage of the LAUPs

S3: Statistical tests for 21 typical mammals

S4: OS\_Ratio in different k and n of Homo sapiens

S5: Workflows for the LAUP analysis

S6: LAUPs number of 24 specie's whole genomes

S7: The occurrences number of CpG-containing sequences compare between

Observe and Expected for Homo sapiens data.

S8: 586 common LAUPs for the whole genome of 21 typical mammalian species

S9: JBLA code and important process algorithms

**Supplementary tables**

**Table 1:** Sequence data source, it is noted that the sequences of 24 representative species are from the NCBI sites with the latest version of the GenBank GBFF (GenBank Flat File) format, which are divided into five lineages, such as Bacteria, Plant, Human, Primates (other than human) and Mammals (excluding Primates).

|  |  |  |  |
| --- | --- | --- | --- |
| Classification | Species | Data size  (Gb) | Version |
| Bacteria | 1. Bacteria | 5.1 | All in ftp://ftp.ncbi.nih.gov/genomes/  archive/old\_refseq/Bacteria/ |
| Plant | 1. Oryza sativa | 0.38 | GCF\_000005425.2\_Build\_4.0 |
| 1. Arabidopsis   thaliana | 0.12 | GCF\_000001735.3\_TAIR10 |
| Human | 1. Homo sapiens | 3.1 | GCF\_000001405.36\_GRCh38.p10 |
| Primats  (except human) | 1. Ornithorhynchus anatinus | 2.0 | GCF\_000002275.2\_Ornithorhynchus\_anatinus\_5.0.1 |
| 1. Pan troglodytes | 3.23 | GCF\_000001515.7\_Pan\_tro\_3.0 |
| 1. Macaca mulatta | 3.24 | GCF\_000772875.2\_Mmul\_8.0.1 |
| 1. Papio anubis | 2.95 | GCF\_000264685.2\_Panu\_2.0 |
| 1. Colobus angolensis | 2.97 | GCF\_000951035.1\_Cang.pa\_1.0 |
| 1. Owl monkey | 2.93 | GCF\_000952055.1\_Anan\_1.0 |
| 1. Callithrix   jacchus | 2.91 | GCF\_000004665.1\_Callithrix\_jacchus-3.2 |
| 1. Microcebus murinus | 2.49 | GCF\_000165445.2\_Mmur\_3.0 |
| 1. Otolemur garnettii | 2.52 | GCF\_000181295.1\_OtoGar3 |
| Mammals  (except Primates) | 1. Rattus   norvegicus | 2.87 | GCF\_000001895.5\_Rnor\_6.0 |
| 1. Mus   musculus | 2.81 | GCF\_000001635.25\_GRCm38.p5 |
| 1. Oryctolagus   cuniculus | 2.74 | GCF\_000003625.3\_OryCun2.0 |
| 1. Canislupus familiaris | 2.41 | GCF\_000002285.3\_CanFam3.1 |
| 1. Miniopterus   natalensis | 1.8 | GCF\_001595765.1\_Mnat.v1 |
| 1. Tupaia   chinensis | 2.85 | GCF\_000334495.1\_TupChi\_1.0 |
| 1. Erinaceus   europaeus | 2.72 | GCF\_000296755.1\_EriEur2.0 |
| 1. Dasypus novemcinctus | 3.63 | GCF\_000208655.1\_Dasnov3.0 |
| 1. Loxodonta   africana | 3.2 | GCF\_000001905.1\_Loxafr3.0 |
| 1. Echinops   telfairi | 2.95 | GCF\_000313985.1\_EchTel2.0 |
| 1. Monodelphis domestica | 3.6 | GCF\_000002295.2\_MonDom5 |

**Table 2:** CG and AG (purine) content percentage of the LAUPs in the WG. The “\*” indicates that there is no LAUP existing for the specie.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Species | K=10 | | K=11 | | K=12 | | K=13 | | K=14 | |
| **CG** | **AG** | **CG** | **AG** | **CG** | **AG** | **CG** | **AG** | **CG** | **AG** |
| Bacteria | \* | \* | \* | \* | \* | \* | \* | \* | 50.89% | 49.91% |
| *Homo sapiens* | \* | \* | 61.33% | 50.07% | 59.26% | 50.03% | 58.53% | 50.01% | 56.22% | 50% |
| Plants | \* | \* | \* | \* | 62.72% | 50% | 59.28% | 50% | 55.71% | 50% |
| Primates | \* | \* | \* | \* | \* | \* | 59.07% | 50% | 57.94% | 50% |
| Mammals | \* | \* | \* | \* | \* | \* | 58.78% | 50% | 57.99% | 50% |

**Table 3:** Statistical tests for 21 typical mammals. Once the P value is less than 0.05, we use “Yes” to indicate statistically significance, otherwise “No”.

|  |  |  |  |
| --- | --- | --- | --- |
| Species | First occurred LAUP number | AT/CG | AG/CT |
| 1. Homo sapiens | 100 | Yes | No |
| 2. Pan troglodytes | 795 | Yes | No |
| 3. Macaca mulatta | 440 | Yes | No |
| 4. Papio anubis | 790 | Yes | No |
| 5. Colobus angolensis | 1143 | Yes | No |
| 6.Owl monkey | 1095 | Yes | No |
| 7. Callithrix jacchus | 664 | Yes | No |
| 8. Microcebus murinus | 70 | Yes | No |
| 9. Otolemur garnettii | 1307 | Yes | No |
| 10. Rattus norvegicus | 253 | Yes | No |
| 11. Mus musculus | 1 | No | No |
| 12. Oryctolagus cuniculus | 253 | Yes | No |
| 13. Canis lupus familiaris | 421 | Yes | No |
| 14. Miniopterus natalensis | 315 | Yes | No |
| 15. Tupaia chinensis | 222 | Yes | No |
| 16. Erinaceus europaeus | 3 | No | No |
| 17. Dasypus novemcinctus | 283 | Yes | No |
| 18. Loxodonta africana | 321 | Yes | No |
| 19. Echinops telfairi | 87 | Yes | Yes |
| 20. Monodelphis domestica | 2 | No | No |
| 21. Ornithorhynchus anatinus | 3 | No | Yes |

**Table 4** OS\_Ratio in different k and n of Homo sapiens. OS\_Ratio describe the difference between Observe (k, n) and Expect (k, n).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CG number | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| k=4 | 0.6431 | 0.2543 |  |  |  |  |  |
| k=5 | 0.6673 | 0.2357 |  |  |  |  |  |
| k=6 | 0.6937 | 0.2407 | 0.2587 |  |  |  |  |
| k=7 | 0.6623 | 0.1699 | 0.0963 |  |  |  |  |
| k=8 | 0.7068 | 0.1833 | 0.1299 | 0.1511 |  |  |  |
| k=9 | 0.7513 | 0.1951 | 0.1418 | 0.1744 |  |  |  |
| k=10 | 0.7953 | 0.2076 | 0.1505 | 0.1828 | 0.7106 |  |  |
| k=11 | 0.8636 | 0.2570 | 0.2211 | 0.3905 | 1.8491 |  |  |
| k=12 | 0.9034 | 0.2644 | 0.2124 | 0.3622 | 1.0635 | 32.2200 |  |
| k=13 | 0.9463 | 0.2714 | 0.2047 | 0.3316 | 0.8434 | 10.1374 |  |
| k=14 | 0.9922 | 0.2773 | 0.2018 | 0.3056 | 0.7851 | 4.0734 | 255.6871 |

**Supplementary figures**

**Figure 1:** (A) Workflow for the LAUPs counting procedure; (B)Workflow for the statistical test; (C)Workflow for the CpG-containing sequences analysis.

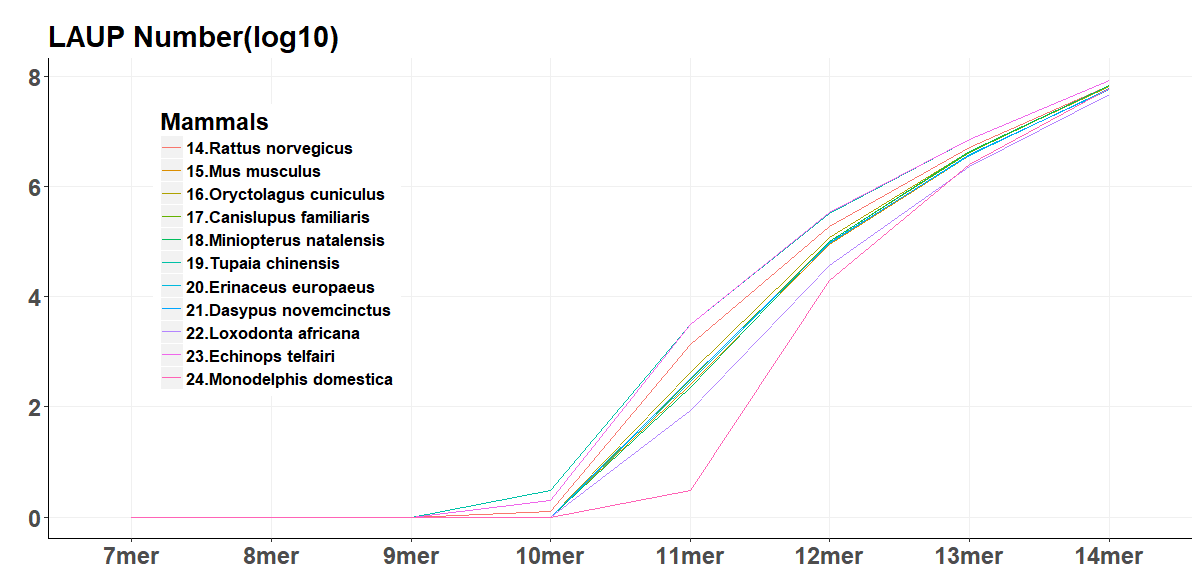
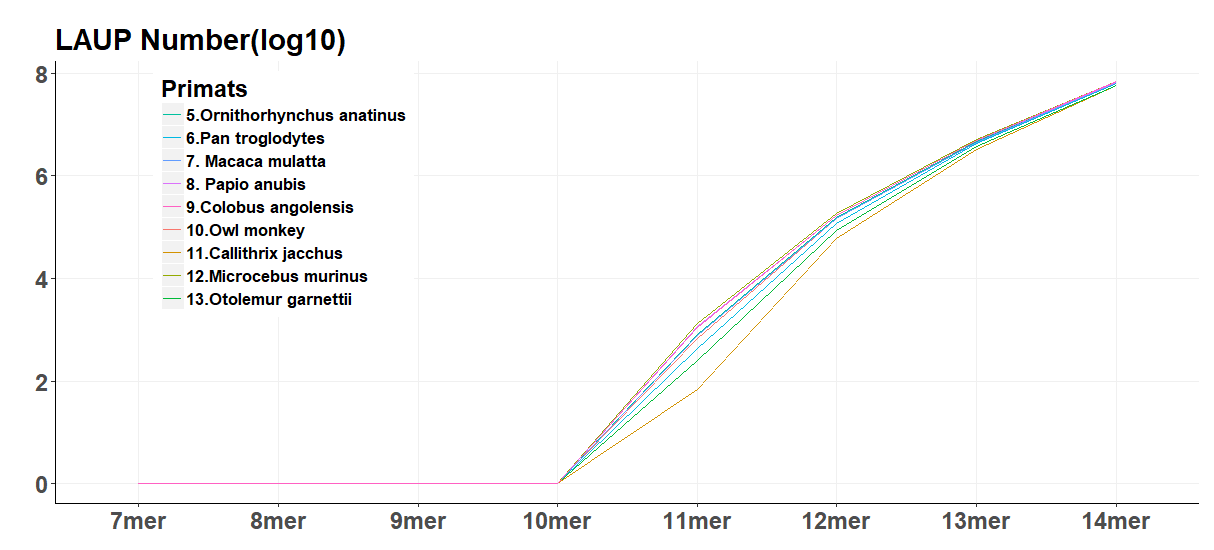
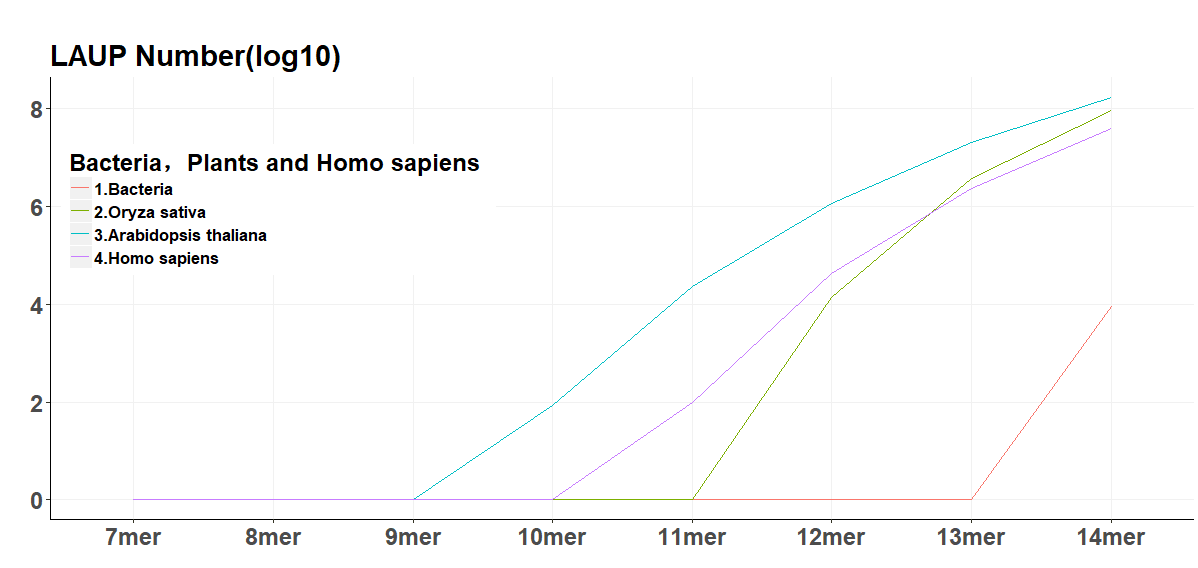




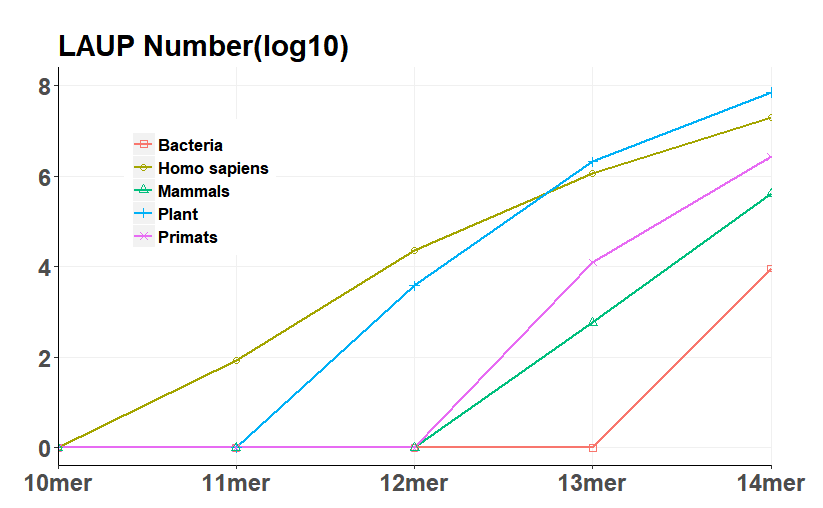


**Figure 2:** Lineage-associated never represented permutations (LAUPs) number of 24 specie's whole genome (WG) (A).The number (B) and proportion (C) of all permutations (4k) of first appeared common LAUPs is generally low for WG, here common LAUPs means those LAUPs which spices in the same classification all have. Horizontal axis represents the length of K-mer. The vertical axis employs the logarithmic function to represent the number (A&B) or the proportion (C) of LAUPs for each length of K-mer.

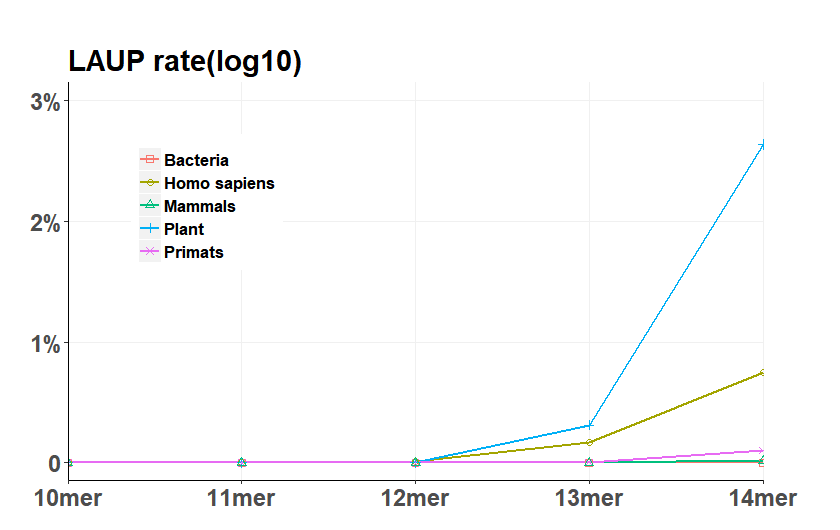
**A.**



**B．**

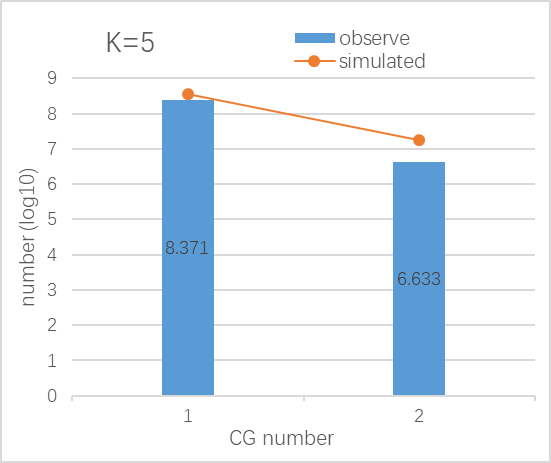
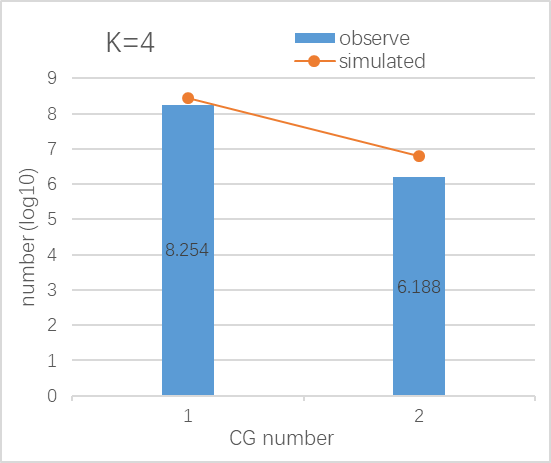


**C.**

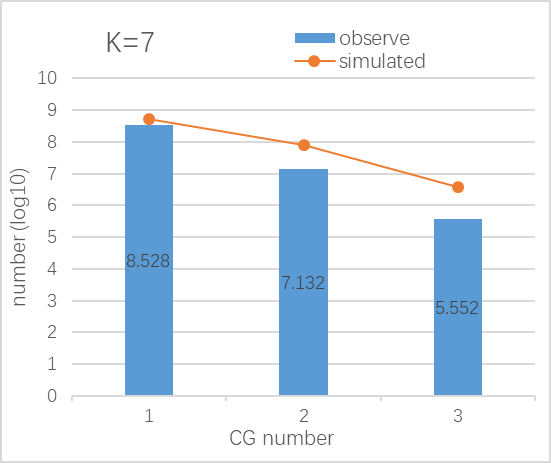
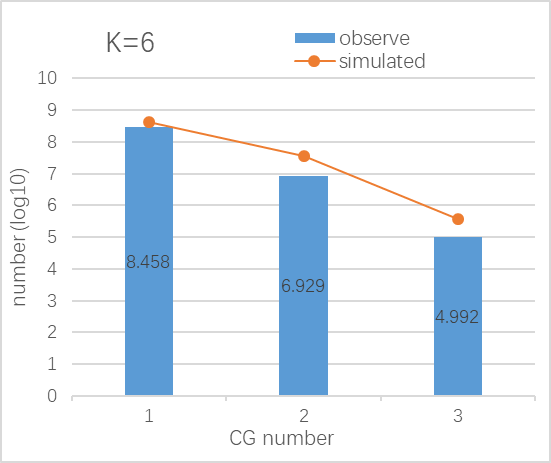


**Figure 3:** The occurrences number of CpG-containing sequences compare between Observe and Expected for Homo sapiens data. The horizontal and vertical axes represent CG number and the occurrences number of CpG-containing sequences.

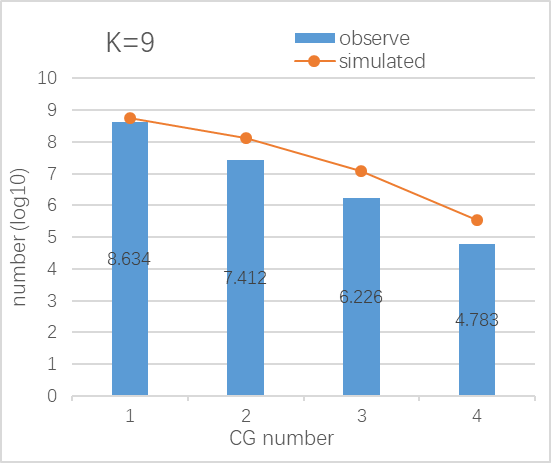
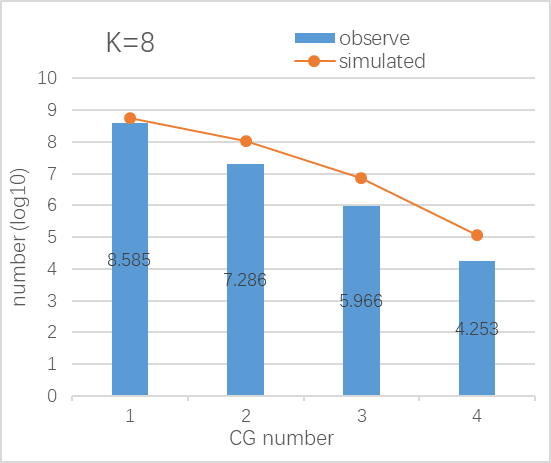
A.  B.



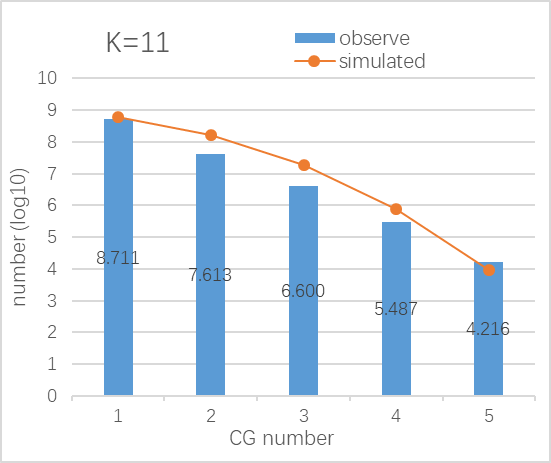
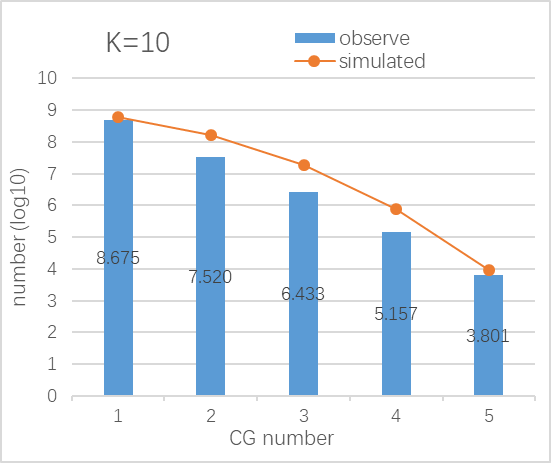
C. D.



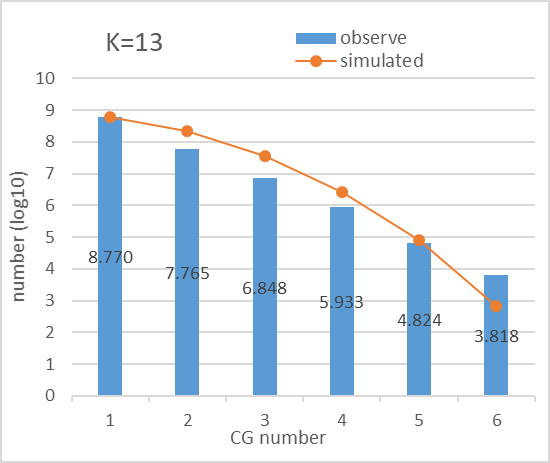
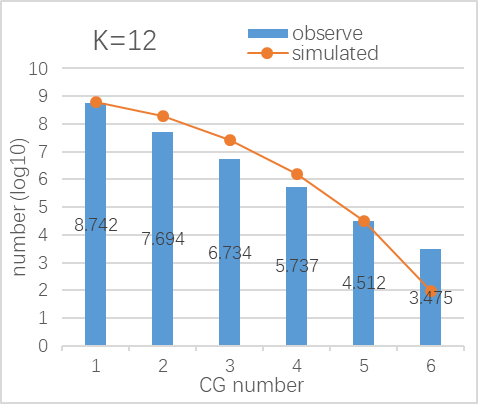
E. F.



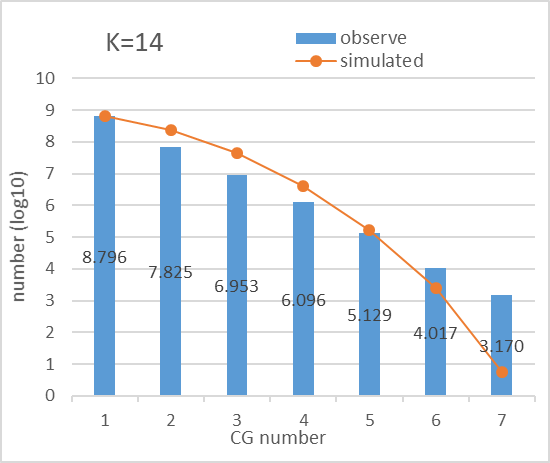
G. H.



I. J.



K.



**Supplementary data**

# 586 common LAUPs for the whole genome of 21 typical mammalian species

CGATTGCGCGAAA

ACTTCGCGCGAAA

TACTCGCGCGAAA

GCGTTCGCGATAA

GATCGCGCGATAA

ATCGCGCGACTAA

CGTCGATGCGTAA

ACGACGCGCGTAA

TGATCGCGCGTAA

ATCGCGCGCGTAA

TCGTACGCCGCAA

TATTCGCGCGCAA

GTCGCGTAACGAA

CTATCGCGACGAA

ATACGCGCTCGAA

CGTAACGGTCGAA

CGTACCGGTCGAA

GTTACGACGCGAA

TATCGGTCGCGAA

GACGATCCGCGAA

TATACGCGCGGAA

TACGCGCGTAATA

TCGCGCACGAATA

CGAGCGACGAATA

CGTCGCTCGAATA

TCGAGCGCGAATA

CTACGCGCGAATA

TTGCGCGCGAATA

CCGCGACGATATA

ATCGCGCGTTATA

CGCGTCGAGTATA

CGGTCGACGTATA

CGCTCGTCGTATA

CTCGACGCGTATA

CCGCTCGCGTATA

CTACGCGCGTATA

TTCCGCGCGTATA

CCGACGGCGTATA

TCGACGCGACATA

CGATCGCGACATA

TACGCGCGTCATA

TCGTCGCGCCATA

TCGTACGCGCATA

CGGCGCGTTGATA

CTCGCGAACGATA

CGCGACTACGATA

CGAAGCGACGATA

TCTCGCGACGATA

TCGCGCGACGATA

TTCGCGACCGATA

ACGCGGACCGATA

CGGTACGCCGATA

CGATTACGCGATA

CCGGTACGCGATA

CGTCGACGCGATA

TCGAATCGCGATA

CTACGTCGCGATA

GTTCGTCGCGATA

GCGTACCGCGATA

GTCGTCCGCGATA

ACGGTCCGCGATA

GTAACGCGCGATA

CCATCGCGCGATA

TGATCGCGCGATA

CCGTCGCGCGATA

GCGTCGCGCGATA

TGGTCGCGCGATA

CGTACGGGCGATA

CGCGGACGTATTA

TACGCGCGCATTA

CGCGCGACGATTA

CGTATCGCGATTA

GTACGCGCGATTA

TGTCGCGCGATTA

TCGCGCGGATTTA

ACGTTCGCGCTTA

ATACGCGCGGTTA

CGCGTCGCAACTA

CGACGCGATACTA

CGACCGCGTACTA

ATCGCGCGTACTA

GTCGCGCGTACTA

GTCGTCGCGACTA

CGTCGGTACGCTA

TCGCGGTACGCTA

ACGCGTCGAAGTA

CGTCGCGCTAGTA

ATCGTCGCGAGTA

TTTCGCGCGAGTA

ATCCGCGCGAGTA

GCGCGTAAACGTA

ATCGACGAACGTA

GGCGACGAACGTA

GCCGTCGAACGTA

CCGCGACTACGTA

GTCGCACGACGTA

TCGCATCGACGTA

ACTACGCGACGTA

ATTGCGCGACGTA

CGCGCGAATCGTA

GCGCGATATCGTA

GCGCGCGATCGTA

CGAACGATTCGTA

CGCGCGATTCGTA

CGCGAATTTCGTA

TCGCGAAGTCGTA

TCGGTTCGCCGTA

CGCGGATAGCGTA

TCGTACGAGCGTA

CGCGTAATGCGTA

CCGTCGATGCGTA

CGCGTTACGCGTA

TCGTAATCGCGTA

GTACGATCGCGTA

ACGCGATCGCGTA

CGCTAGTCGCGTA

GATACGTCGCGTA

CGATACGCGCGTA

TATTACGCGCGTA

TATGACGCGCGTA

GTTGTCGCGCGTA

CTAACCGCGCGTA

TCGACCGCGCGTA

TAATGCGCGCGTA

AATCGCGGGCGTA

GCGCGAAACGGTA

CGACGCGTCGGTA

CGTCGACCGTACA

CGTATCGCGTACA

CAATTGCGCGACA

TAATCGCGCGACA

TTACGCGCGATCA

TATCGCGCGATCA

TATCGCGCGACCA

AATCGACCGCGCA

TCGGTACGCGAGA

TATCGTCGCGAGA

TCGCGTCGAACGA

CGATCGCGAACGA

GGCGCGTATACGA

TCGCACGATACGA

ACGCGCGATACGA

TACGCGATTACGA

CTCGCGATTACGA

CGCGGATTTACGA

TACGCTCGTACGA

GATACGCGTACGA

GTTTCGCGTACGA

ACGTCGCGTACGA

CTACCGCGTACGA

TATGCGCGTACGA

GTCGCGCGTACGA

TTGCGGCGTACGA

ATCGGGCGTACGA

ATACGCGCGACGA

TATGGCGCGACGA

TCGCGCGAATCGA

CGTGACGCATCGA

TATCGCGATTCGA

TATTCGCGCTCGA

CGCGTAACGTCGA

CGGCGATCGTCGA

CGTAGTCCGTCGA

CTTATCGCGTCGA

TATGTCGCGTCGA

TACGCGCGGTCGA

TACGGCGAACCGA

TCTCGCGTACCGA

AACCGCGTAGCGA

CGCGACGATGCGA

TACGTCGATGCGA

CGTCGAATTGCGA

TCGTATCGTGCGA

CGACGTAACGCGA

AGTTCGAACGCGA

GCGTTATACGCGA

CGACCGTACGCGA

GATCGGTACGCGA

TCGTTCGACGCGA

CGTCGAATCGCGA

CGACAATTCGCGA

CGCGAATTCGCGA

TACGACTTCGCGA

CGATAGTTCGCGA

ATATCGTTCGCGA

ATACGACTCGCGA

AGTATCGTCGCGA

TAGCGTACCGCGA

CGATTATGCGCGA

TATTCGTGCGCGA

CTTATACGCGCGA

ATAGTACGCGCGA

GTCGTACGCGCGA

CACTATCGCGCGA

ACCTATCGCGCGA

TCGATTCGCGCGA

TATCGTCGCGCGA

TAAATCCGCGCGA

CAATTGCGCGCGA

CAATTCGCGCGGA

CGTATACGGCGGA

AATCGCGCGAAAT

CGTCGCGACTAAT

TACGTCGCGCAAT

GTTCGCGCGCAAT

CGCGTACGCGAAT

GGTTCGCGCGAAT

CGACCGCGAATAT

ACGGCGCGAATAT

CGACGCGTTATAT

CGTCGACGCATAT

TCGCGAACGATAT

CGCGACGCGATAT

CGTACCGCGATAT

GGTCGCGCGATAT

CCGCGCGAATTAT

GTCGCGCGATTAT

TCGCGCGTACTAT

CGCGACGAAGTAT

TCGCGAGTCGTAT

CGCGCTAGCGTAT

TTCGAGCGCGTAT

TCGTCGCGCGTAT

TAACCGCGCGTAT

CGTACGGGCGTAT

GTCGACGCGGTAT

CGCGACGAACGAT

CGCGACGCACGAT

TACTCGCGACGAT

CGCGAACGTCGAT

TACGTTCGTCGAT

GTCGCGTACCGAT

CGCGTTATCCGAT

TCGTACGCCCGAT

CGAACTCGCCGAT

CGCATTACGCGAT

TATAACGCGCGAT

GTATACGCGCGAT

TAGTACGCGCGAT

CGATTCGCGCGAT

CATTTCGCGCGAT

TTAGTCGCGCGAT

AACGTCGCGCGAT

TTACGCGCGCGAT

CGTATCCGGCGAT

TACTCGCGCGGAT

CGCGTACCGAATT

CCGTACGCGAATT

CCGCGACGATATT

CGTCGCGCGTATT

CGCGCGAACGATT

GCGATCGTCGATT

TGCGCGGTCGATT

TACGCCCGCGATT

ATTTCGCGCGATT

ATCGCGCGACGTT

GTATTCGCGCGTT

TCGCTACGCGGTT

GTCGCGCGAAACT

CGGTCGACGAACT

TCGCGTTCGAACT

CGCGCCGTATACT

TCGCGACGATACT

CGCGCCGATAAGT

TTTCGCGCGAAGT

ACGTCGCGATAGT

GACGCGCGATAGT

TACGTCGCGTAGT

CGCGACGTATTGT

TAAGCGCGAACGT

TCGTACGCGACGT

ACTATCGCGACGT

CGCGATATATCGT

CGCGAATTATCGT

CGCAACTTATCGT

CGCTATACGTCGT

GCGGATACGTCGT

TTACGCGCGTCGT

TATCGCGGACCGT

GTACGCGATCCGT

ATATTCGCGCCGT

GCGCGTATAGCGT

TACTTCGACGCGT

TACGCGATCGCGT

TATCGGTCCGCGT

GGTAATCGCGCGT

TCGTATCGCGCGT

TCGCGCGATAGGT

CGTACGACGAAAC

TCGTACGCGAAAC

CGTCGCGAATAAC

CGCGACCGATAAC

TTCGCGTCGTAAC

TACGCGCGACAAC

TATCGCGACGAAC

GTATCGCGCGAAC

ATTGCGCGCGAAC

CGAACGCGAATAC

AACGCGCGAATAC

CGTCGCGCTATAC

ATCGCGCGTATAC

CGCGCAACGATAC

GGCGCGACGATAC

CGCGGTCCGATAC

CGCGTTGCGATAC

CGAAACGCGATAC

GTTCGCGCGATAC

CGCGGCGAATTAC

TATCGCGCGTTAC

CGCGGTACGCTAC

CGTCGCGATGTAC

CGAATCGACGTAC

TACGCGATCGTAC

CGTAATCGCGTAC

ACGGATCGCGTAC

TAATCGCGCGTAC

CGATACGCGGTAC

GCGCGTATACGAC

TCGCGCGTACGAC

TAGTCGCGACGAC

TATCGCGGACGAC

CGCGATAATCGAC

CGATATCGTCGAC

ATACCGCGTCGAC

CGCGTATTGCGAC

TACGTCGTGCGAC

CCGTATACGCGAC

CCGATTACGCGAC

TTCGTTACGCGAC

ATCGGTACGCGAC

CGTTATTCGCGAC

GGTACGTCGCGAC

CGTTATGCGCGAC

TAGTACGCGCGAC

TCGTACGCGCGAC

ATAATCGCGCGAC

CTTATCGCGCGAC

AGTTTCGCGCGAC

CGCGACGTATATC

TACGCGACGTATC

TCGTACGCGTATC

CGCGCGAACGATC

TCGCGTACCGATC

CGATTCGCCGATC

TTATCGCGCGATC

TTCGCGGATCGTC

ACTATCGCGCGTC

ATTCGCGCGAACC

ACGCGCGATTACC

GTCGCGACGTACC

CGTACGTTCGACC

ATATCGCGCGACC

TACGTTCGTCGCC

TCGTATACGCGCC

GTATCGTCGCGCC

TCGCGTATAACGC

TTATCGCGAACGC

CGCCGATATACGC

TATCGCGGTACGC

TATCGCGCGACGC

CGAACGATATCGC

CGTCGATTATCGC

AATCGACGATCGC

ACGACGTATCCGC

ACGCTATACGCGC

GTCGTATACGCGC

TACGTTTACGCGC

TACGATATCGCGC

CGTAAGATCGCGC

TACCGTTTCGCGC

CGATAAGTCGCGC

TACGATCGCGCGC

CGCGCGAATTGGC

TACGTTCGACGGC

TCGCGCGTATAAG

TCGACGCGATAAG

GTCGCGCGATAAG

CGTACGACGATAG

TTCGTCGCGATAG

TACGCGCGGTTAG

TATCGCGACGTAG

CGCGCGATCGTAG

TATACGCGCGTAG

TATTCGCGCGTAG

TCGTACGCGGTAG

TATACGCGTCGAG

TCGTAATCGCGAG

TATCGTTCGCGAG

ATCGCGCGAAATG

CGCGCGAAATATG

CGCGAATCGTATG

TGTCGCGCAATTG

TCGCGCGCAATTG

TCCGCGCGAATTG

CCGTCGACGATTG

TCGCGCGATAGTG

CGTATCGCGCGTG

GTCGCGAATAACG

CGGACGTATAACG

GTCGCGCATAACG

CCGGTATCGAACG

CGATACGCGAACG

CGCGCGATATACG

CGACGATTATACG

CGTGCGCTATACG

TCCGCCGTATACG

CGCGCGTGATACG

CCGCAACGATACG

TGTACGCGATACG

TAATCGCGATACG

CACGCGCGATACG

ATCGCCGGATACG

GTACGCGATTACG

GCGCGATCTTACG

TTCGACCGTTACG

TCGACGGACTACG

CGTCGATAGTACG

CGACGCTAGTACG

CGCGAAACGTACG

GGTCGAACGTACG

CTATCGTCGTACG

GTTTCGTCGTACG

ATACGCCCGTACG

TATCGCCCGTACG

TTCGACCGGTACG

ATATCGCGGTACG

TCGATGCGTCACG

CGTATAGCGCACG

CCGTATCGAGACG

CCGATATACGACG

GCGATAATCGACG

CGTACTATCGACG

TTACGCATCGACG

TCGCAATTCGACG

TCGCGATTCGACG

CGATAGTTCGACG

ATATGCGTCGACG

TATCGCGTCGACG

TGTACGGTCGACG

TAGCGTACCGACG

TATTCGAGCGACG

CGAGTATGCGACG

GTACATCGCGACG

GTTATTCGCGACG

ATTAGTCGCGACG

GTATAGCGCGACG

TACTAGCGCGACG

AATACGCGCGACG

TCGCGCATAATCG

TATCGCGTAATCG

TTTCGCGCAATCG

ATCGCGCGAATCG

GATCGGCGAATCG

CGCGCCTATATCG

CGCGCGTATATCG

GTCGACGATATCG

CCGCGAATTATCG

CGACGTATTATCG

CCGCGTATTATCG

GCGCGACTTATCG

CGTCGAACTATCG

TCGCGAACTATCG

CGTTCGCGTATCG

GTACCGCGTATCG

TACGCGCGTATCG

TCGTTCGCGATCG

TATGTCGCGATCG

CGCGGACTATTCG

CCGGTACGATTCG

GTACGTCGATTCG

GTATCGCGTTTCG

TATCGTCGCTTCG

CGCGATTAGTTCG

ATCGGCGAGTTCG

TACGAATCGTTCG

GCGATATCGTTCG

GTATTCGCGTTCG

CGTCGCATACTCG

TATTCGTCGCTCG

CGACGTATAGTCG

TCGCGAATTGTCG

CGATAATACGTCG

CGACTATACGTCG

TCGCGTTACGTCG

CGTATAATCGTCG

CGTACTAGCGTCG

ATATAACGCGTCG

TACCGACGCGTCG

TAGTATCGCGTCG

TCGCGTACGGTCG

TAGTACGCGGTCG

ATATTCGCGGTCG

CGCATACGGGTCG

CGCGATATAACCG

CGCGAGTATACCG

CCGTCGTATACCG

TATCGGCGTACCG

TATACGTCGACCG

AGTTCGTCGACCG

CGTTATACGTCCG

TCGACGATCGCCG

TATCAACGCGCCG

ACGACGTATAGCG

TACGCGACTAGCG

TATACGACGAGCG

ATCGCGTAATGCG

CGACCCGTATGCG

ACGATAAGTTGCG

ATCGGATAACGCG

TACGCGTAACGCG

GTATCGCAACGCG

CCGTTCGAACGCG

GTCGCAATACGCG

TACGCATTACGCG

TCGACGTTACGCG

ATTCGCGTACGCG

AATTCGGTACGCG

TATACTCGACGCG

TAGTTGCGACGCG

CGAACTAATCGCG

ACGATATATCGCG

CGGTTATATCGCG

GTCGATTATCGCG

TACGAAATTCGCG

ACGATAATTCGCG

TCGCGAATTCGCG

CATACGATTCGCG

CGTACGTTTCGCG

ATCGACGTTCGCG

CGGTATACTCGCG

TATCGTAGTCGCG

ACAATACGTCGCG

GATATACGTCGCG

TCGCATCGTCGCG

ATACTTCGTCGCG

ATCGTTCGTCGCG

ATCGTGCGTCGCG

ATATCGCGTCGCG

GTTATCGGTCGCG

GTAGCGTACCGCG

GTATCGGACCGCG

TCGTAAATCCGCG

TACGCTATCCGCG

CGAATAGTCCGCG

TAATACGTCCGCG

GTAATTCGCCGCG

ATACGCTAGCGCG

GTATCGTTGCGCG

CGATATACGCGCG

CGTATCACGCGCG

TACGAATCGCGCG

CGTATATCGCGCG

CTACGATCGCGCG

GCCAATTCGCGCG

TACGATTCGCGCG

CATATTTCGCGCG

AATCGTTCGCGCG

GATCGTTCGCGCG

TAATCGTCGCGCG

CGATATAGGCGCG

AGTATACGGCGCG

ACTTATCGGCGCG

GCGTATATCGGCG

TATCGCGCGATGG

CGCGTTCGAACGG

GTCGCGTATACGG

CGTCTCGATACGG

AATTCGCGTACGG

CGGTATACGACGG

TACGCATCGACGG

CAATCGTCGACGG

TATCGCGCGACGG

GTCGCGTAATCGG

CGTCGTATATCGG

TATACGCCGTCGG

CGTTCGATACCGG

CGAATCGTACCGG

TATCGCGTACCGG

TATACGCGAGCGG

CGTATCGTTGCGG

CGATAATACGCGG

CGATAATTCGCGG

TACGTAGTCGCGG

AATATCGTCGCGG

TATATCGTCGCGG

ATAATTCGCGCGG

**Supplementary methods**

1. We develop a Jellyfish based LAUP analysis software (JBLA) by integrating Jellyfish, MEME and NCBI genome database applications for efficient LAUP sequence analysis. Here we list some important process algorithms in the software and the main code of JBLA is in the file JBLA\_code.zip.
   1. **LAUP counting process algorithm**

**LAUP counting process algorithm（Input:** K and P are index; **Outpu**t: LR for LAUP**）**

1: **For** K = 1,…,N (N is the upper K limit)

2: Initialize two Jellyfish parameter: K value and hash table size for hash()

3: Send each K-mer result into hash(K) .

4: **For** P = 1,…K4,

5: Assign each permutation of A, T, C and G a numerical index P

6: String arr[] = { "A", "T", "C", "G" }

7: String kmer\_string;

8: **For** i = 0,…K-1,

9: kmer\_string += arr[((P / pow(4, i))) % 4] ;

10: **End For**

11: **If** the permutations kmer\_string is in hash(K):

12: Put this permutation kmer\_string into LR.

13: **End If**

14:  **End For**

15:  **End For**

* 1. **Statistical test process algorithm**

**Statistical test process algorithm（Input:** K and N is the index and limit; **Outpu**t: P-value)

1: **For** K = 1, … N,

2: **If** the data follow the normal distribution:

3: Homogeneity test of variance.

4: Use T-test to obtain the P-value for the data.

5: **Else**

6: Use Wilcoxon rank sum test to obtain the P-value.

7: **End If**

8: **If** P-value <= 0.05

9: The data is statistical significance.

10: **Else** The data is not statistical significance.

11: **End If**

12: **End For**

* 1. **CpG-containing sequence Observe counting process algorithm**

**CpG-containing sequence Observe counting process algorithm**

**（Input:** K and i,n is index**; Output:** Observe (k, n)**)**

1: map<int, long long> map1;

2: map<String,long long> map2;

3: String arr[] = { "A", "T", "C", "G" }

4: String kmer\_string;

5: map<String,long long>::iterator it

6: **While**(scanf ( kmer\_string, frequency) !=EOF)

7: Insert <kmer\_string,frequency> to map2;

8: **End While**

9: **For** it=map2.begin…map2.end-1

10: **If**  map2->first[i i+1] = “CG”

11: num++;

12: **End If**

13: map1[num]+=map2->second;

14: **End For**

15: **For** m=1…K/2

16: Observe = map1[m]->second

17:  **End For**

* 1. **CpG-containing sequence Expect Rate counting process algorithm**

**CpG-containing sequence Expect Rate counting process algorithm**

**（Input:** K and i,n is index**; Output:** **Expect \_rate**(k, n)**)**

1: map<int, long long> map1;

2: String arr[] = { "A", "T", "C", "G" }

3: String kmer\_string;

4: **For** i=0,…4^K-1,

5: kmer\_string += arr[((P / pow(4, i))) % 4] ;

6: **For** j=0…K-1

7: **If**  kmer\_string[i j+1] = “CG”

8: num++;

9: **End If**

10: **End For**

11: map1[num]++

12: **End For**

13: **For** n=1…K/2

14: Expect \_rate = map1[n]->second / pow(4,K)

15: **End For**