

基于 RNA-Seq 数据集的转录组从头拼接算法

武思文 李 静 张少强

(天津师范大学计算机与信息工程学院 天津 300387)

摘 要 转录组拼接是基因组测序与功能注解问题的一个重要组成部分。为了提高转录组拼接的精度和效率,文中提出了一种新的转录组从头拼接算法 StepLink。该算法的主要创新点是提出了最左 k-mer(长度为 k 的短序)和右 k-mer 的概念,并运用双重哈希表来存储相邻的每对 k-mer,使得拼接更加迅速、准确。应用该算法对 SRA 数据库中人、狗和老鼠的测序数据分别进行拼接,结果表明该算法比其他已有算法更高效。

关键词 转录组, RNA-Seq, K-mer, 从头拼接算法

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.12.049

De Novo Transcriptome Assembly Algorithm Based on RNA-Seq Datasets

WU Si-wen LI Jing ZHANG Shao-qiang

(College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China)

Abstract Transcriptome assembly is an important part of genome sequencing and function annotations. To improve the precision and efficiency of transcriptome assembly, this paper presented a new de novo transcriptome assembly algorithm called StepLink. The main innovations of this algorithm are presenting two concepts, namely leftmost k-mer (short sequence of length k) and right k-mer, and using the hash of hashes table to store the k-mer pairs, which makes the assembly more quickly and accurately. This algorithm was used to assemble the datasets of human, dog and mouse in the SRA databases respectively. The experimental results suggest that the proposed algorithm has higher efficiency than other existing algorithms.

Keywords Transcriptome, RNA-Seq, K-mer, De novo assembly algorithm

1 引言

转录本是由一条基因通过转录形成的一种或多种可供编码蛋白质的成熟的 mRNA,转录组是某个物种或者特定细胞类型产生的所有转录本的集合。随着第二代测序技术的不断发展,特别是 RNA-Seq 技术^[1]越来越成熟, RNA-Seq 实验提供了大量的转录组短读序(reads),其拼接成的转录组^[2]被广泛用于比较样品基因表达水平的差异。转录本预测工作可以给疾病发病机理、药物开发等方面的研究提供巨大帮助,因此对转录组的研究具有非常现实且重要的意义。

目前,针对转录组拼接的算法主要分为两种:基于参考基因组的拼接算法和从头开始(de novo)的拼接算法^[3]。基于参考基因组的拼接算法主要有 Cufflinks^[4], Scallop^[5], TransComb^[6], StringTie^[7], Bayesemba^[8]等,这类算法要求所要拼接的转录组存在对应的参考基因组。这类算法一般分为 3 步:1)将 RNA-Seq 数据以短读序的方式映射到相应的参考基因组上(称之为 Mapping);2)以短读序在参考基因组上的映射信息为依据构造图;3)对所得图进行遍历获得所有的转录本。基于参考基因组的拼接算法表现很好,但是由于很多物

种的基因组序列尚未被完整测序,因此需要对从头开始的拼接算法进行研究。当前已有的从头拼接算法主要包括 Oases^[9], SOAPdenovo-Trans^[10], IDBA-Tran^[11], Trinity^[12], Bridger^[13-14]等。其中大部分转录本从头拼接算法基于基因组拼接算法的思想,这类算法对于低等生物有很好的拼接效果,但是对于高等生物效果不佳。此外,已有的从头拼接算法大都需要构造复杂的 de bruijn 图,该图的规模非常大,导致算法的时间复杂度非常高。本研究放弃 de bruijn 图,针对缺少参考基因组的生物体第二代高通量测序 RNA-Seq 实验数据,设计了一种全新的转录组从头拼接算法 StepLink^[1]。将该算法与其他同类算法进行对比分析,结果表明:StepLink 针对不同的数据均有更好的鲁棒性,并且在时间复杂度与准确性等方面比已有算法更优。本文第 2 节介绍数据的收集整理;第 3 节介绍算法及其流程;第 4 节对结果进行分析;最后总结全文。

2 数据来源与预处理

2.1 获取序列数据

为了验证所提算法的灵敏性与准确性,方便与其他同类算

¹⁾ <http://www.escience.cn/people/sqzhang/program.html>

到稿日期:2017-10-30 返修日期:2018-01-16 本文受国家自然科学基金(61572358),天津自然科学基金(16JCYBJC23600)资助。

武思文(1994—),女,硕士生,CCF 会员,主要研究方向为生物信息计算;李 静(1990—),女,硕士,主要研究方向为生物信息计算;张少强(1976—),男,教授,主要研究方向为生物信息计算, E-mail: zhangshaoqiang@tjnu.edu.cn(通信作者)。

法进行比较,本研究选取了 SRA 数据库^[15]中的人、狗和老鼠的 RNA-Seq 测序数据作为实验数据,对应数据编号分别是:人(SRX011545 和 SRX011546)、狗(SRX295056)、老鼠(SRX062280)。美国国家生物技术信息中心(NCBI)创建了 SRA 来帮助用户管理测序数据的数据库,并将数据免费提供给科研学者使用。

2.2 数据预处理

本研究在实验过程中使用的数据格式是 FASTQ 格式,因此需要对从 SRA 数据库中下载的测序数据进行预处理。在数据预处理的过程中主要用到两个工具:prefetch 和 fastq-dump。首先,通过 prefetch 命令下载 SRA 格式数据,之后使用 fastq-dump 将数据格式转换为所需的 FASTQ 格式。

3 方法和步骤

本算法的主体设计思路是:首先,使用双重哈希表存储 RNA-Seq 读序;然后,找到最左 k-mer 集合,分别从每个最左 k-mer 开始扩展延伸序列,得到所有叠连群(contig);接着,基于叠连群序列生成粘贴群(affix)集合;最后,通过拼接叠连群和粘贴群,得到最终的转录组。该算法的基本处理流程如图 1 所示。

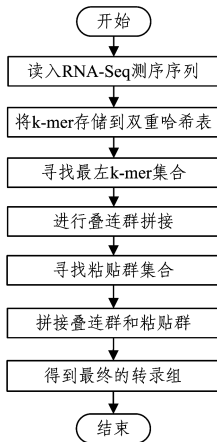


图 1 算法设计流程图

Fig. 1 Flowchart of algorithm design

3.1 构建双重哈希表

首先,将测得的每条读序(长度一般为 50~200 个碱基)划分为更短的 k 长度片段(k 一般取 25~30),称这些 k 长度片段为 k-mer。为了高效地存储所有的 k-mer,本研究使用了一种特殊的数据存储结构——双重哈希表(hash of hashes table)。在使用双重哈希表存储 k-mer 时,以相邻的一对 k-mer (称为邻接 k-mer 对)作为双键,以该邻接 k-mer 对出现的次数作为键值进行存储。在存储完毕后,首先找出转录本起始的 k-mer,再在双重哈希表中快速找到它的邻接 k-mer,然后以获取的邻接 k-mer 为基准,继续寻找它的邻接 k-mer,重复这个过程直到找不到邻接 k-mer,这样就可以基于双重哈希表直接将读序拼接成一条更长的序列。理论上以这种拼接方法取代 de bruijn 图能够大幅缩短拼接时间。

在构建双重哈希表时,需要依次遍历给定的 RNA-Seq 的所有读序,并逐次将出现过的邻接 k-mer 对全部记录到双重哈希表中。具体步骤是:对于每一条读序,以 $2k$ 个碱基长度为窗口,从左到右以步长为 1 进行遍历滑动。在每个滑动窗

口内,称前 k 个碱基序列为左 k-mer,称后 k 个碱基序列为右 k-mer,左 k-mer 和右 k-mer 共同构成邻接 k-mer 对,并以键对的形式放入双重哈希表中,同时累计每对邻接 k-mer 对的出现次数,并将之作为哈希表的键值。与此同时,将每个读序的第一个 k-mer (称为首 k-mer)和它所有的右 k-mer 分别用两个哈希表记录下来(记为 first_kmer_hash 和 right_kmer_hash)。图 2 的示例显示了双重哈希表的构造过程。

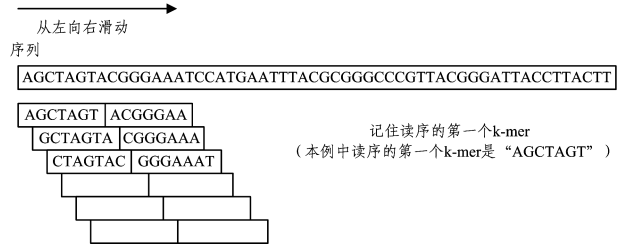


图 2 双重哈希表的构造过程示例

Fig. 2 Example of construction of hash of hashes table

3.2 寻找最左 k-mer 集合

在拼接之前需要寻找每条转录本的拼接起点,为此,本文提出了最左 k-mer 的概念,即在双重哈希表中不以右 k-mer 的形式存在的首 k-mer。所有读序的首 k-mer 和右 k-mer 已经分别用两个哈希表进行了记录,找出存在于 first_kmer_hash 且不存在于 right_kmer_hash 的 k-mer 集合,就是所有的最左 k-mer 集合。

3.3 构造叠连群(contig)

根据最左 k-mer 的定义和其获取过程可知,最左 k-mer 一定是转录本序列的起始短序。对于每一个最左 k-mer,在双重哈希表中选择其出现频率最高的邻接右 k-mer 进行序列扩展,之后将该右 k-mer 作为新的左 k-mer,继续选择其出现频率最高的邻接右 k-mer 进行序列扩展。重复上述过程直到无法扩展。

上述扩展过程的结束并不代表叠连群拼接工作结束,而是需要调整 k-mer 以继续拼接。下面的拼接工作采用一种回溯试探的方案,该方案的具体操作是:在找不到邻接右 k-mer 时,在当前 k-mer 的基础上倒退一个碱基得到新的 k-mer,检查此 k-mer 是否有对应的邻接右 k-mer。若有,则继续拼接;若没有,则继续倒退一个碱基。重复上述过程,直到无法找到对应的邻接右 k-mer,结束回溯(注意:最多倒退 $k-1$ 步)。回溯方案的示例如图 3 所示。以每个最左 k-mer 为起点,通过反复的扩展和回溯构造出最终的叠连群集合。构造结束后从双重哈希表中删除叠连群集合所包含的所有邻接 k-mer 对。



图 3 叠连群的回溯方案

Fig. 3 Backtracking scheme of contigs

3.4 寻找粘贴群(affix)

若此时双重哈希表非空,则继续对双重哈希表剩余的邻接 k-mer 对进行拼接,得到能够附加到叠连群上的序列,并称其为粘贴群(affix)。粘贴群的寻找过程是:以 k 个碱基长度为窗口,依次滑动遍历之前得到的每个叠连群,判断滑动窗口所对应的 k-mer 在当前双重哈希表中是否存在,若存在,则以该 k-mer 为起点,使用构建叠连群的方法寻找粘贴群。在寻

找过程中,每得到一个粘贴群,都从双重哈希表中删除粘贴群所包含的所有邻接 k -mer 对。在遍历完所有叠连群之后,得到第一批粘贴群。将得到的粘贴群作为基准,继续寻找粘贴群的粘贴群。重复这一过程,直到双重哈希表为空。将得到的粘贴群按照产生的先后次序排序。至此,所有 k -mer 全部拼接成了叠连群和粘贴群。图 4 展示了粘贴群的查找过程。

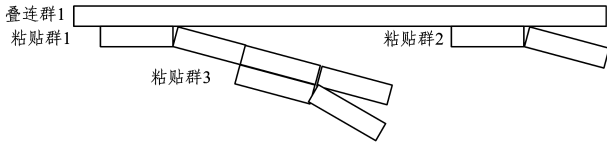


图 4 粘贴群的查找过程

Fig. 4 Discovery process of affixes

3.5 叠连群和粘贴群的拼接

算法最后,按照排好序的粘贴群次序,依次将每个粘贴群粘贴到不断更新的叠连群上。若某个粘贴群的第一个 k -mer 在某叠连群或粘贴群中存在,其最后一个 k -mer 在该叠连群/粘贴群中不存在,则称这种情况为仅首可粘贴;若某个粘贴群的第一个 k -mer 在某叠连群或粘贴群中存在,同时其最后一个 k -mer 在该叠连群或粘贴群中也存在,则称这种情况为首尾可粘贴。

对于每一个初始叠连群,从头到尾依次按粘贴群次序将首尾可粘贴的粘贴群粘贴到叠连群(要求粘贴群不能交叉粘贴),直到遇到第一个仅首可粘贴或者无任何可粘贴为止,形成一个新的叠连群。每个粘贴群在使用后,都从当前的粘贴群集合中删除。以新的叠连群为基准,依次把剩余的粘贴群粘贴到叠连群并更新叠连群集合。重复上述过程,直到粘贴群集合为空,最终得到的每个叠连群即为一个转录本,所有的叠连群形成最终的转录组。拼接示意图如图 5 所示。

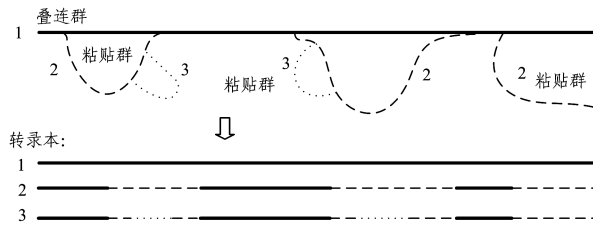


图 5 拼接示意图

Fig. 5 Diagram of assembly

4 数据结果分析

目前,对转录组拼接结果的评价还没有一个系统的方法与标准,但可以根据现有的转录组拼接算法对拼接所得的转录组给出合理的评价。在此,称拼接所得的转录本为候选转录本,已知的转录本为参考转录本。若所得的候选转录本有 95% 以上与参考转录本匹配,则认为此次拼接是成功的。

本研究从敏感度(sensitivity)和准确度(specificity)这两个方面对算法的拼接效果进行评价分析。 S 代表敏感度, A 代表准确度。

敏感度可表示为 $S = \frac{m}{M}$,准确度可表示为 $A = \frac{m}{N}$ 。其中, m 代表拼接所得的真阳性转录本数量, M 代表真实转录组所有的转录本数量, N 代表拼接所得的转录本数量。

4.1 不同算法的参数设置

为了证明 StepLink 的高效性,本文将 StepLink 与其他从头拼接算法进行了比对分析。Cufflinks 是基于参考基因组的拼接算法,在此作为一个评价从头开始的拼接算法的基准,也列入比对范围。不同算法使用不同参数会影响最终的拼接效果,在此 k -mer 长度的选取对结果的影响较大。各个算法的 k 值设置如表 1 所列,未指名的 SOAPdenovo-Trans 和 Cufflinks 的参数采用默认值。

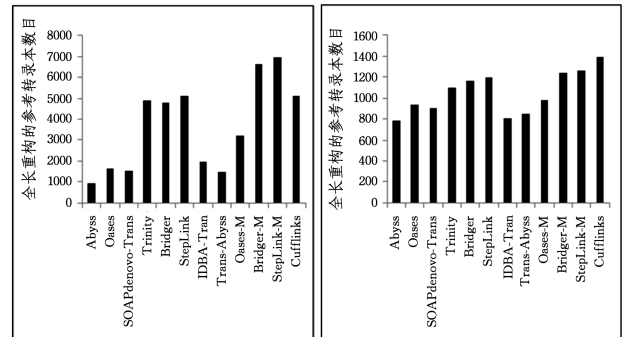
表 1 参数设置

Table 1 Setting of parameter

	人类	狗	老鼠
Abyss	25	25	33
Oases	25	25	31
Trinity	25	25	25
Bridger	25	25	31
StepLink	25	25	31
IDBA-Tran	21,26,33,37,43	21,26,33,37,43	25,31,37,43,49
Trans-Abyss	21,26,33,37,43	21,25,33,37,43	25,31,37,43,49
Oases-M	21,26,33,37,43	21,25,33,37,43	25,31,37,43,49
Bridger-M	21,23,25,27,29	21,23,26,27,29	23,25,27,29,31
StepLink-M	21,26,33,37,43	21,25,33,37,43	25,31,37,43,49

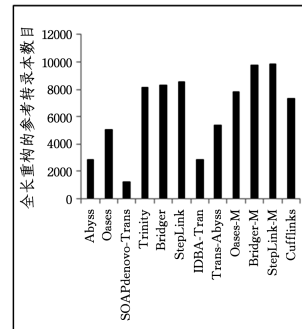
4.2 敏感度和精确度分析

在敏感度方面,StepLink 的表现很好(见图 6),优于 Trinity 和 Bridger。尤其是在人和老鼠的 RNA 测试数据方面,StepLink 的表现比 Cufflinks 更好。在人的 RNA 测序数据上,StepLink-M 的效果有很大的提升,说明针对人的 RNA 数据, k 值的选择至关重要。



(a) 人类的测序数据

(b) 狗的测序数据



(c) 老鼠的测序数据

图 6 对 3 种 RNA 测序数据采用不同算法所得的敏感度对比

Fig. 6 Sensitivity comparison of different algorithms for three RNA sequencing datasets

在精确度方面,StepLink 的表现同样很好(见图 7)。从图 7 可以看出,StepLink 在精确度方面基本不输于其他算法,尤其是在人的 RNA 测序数据方面,StepLink 的表现比 Bridger 更好,且远胜于 Cufflinks。

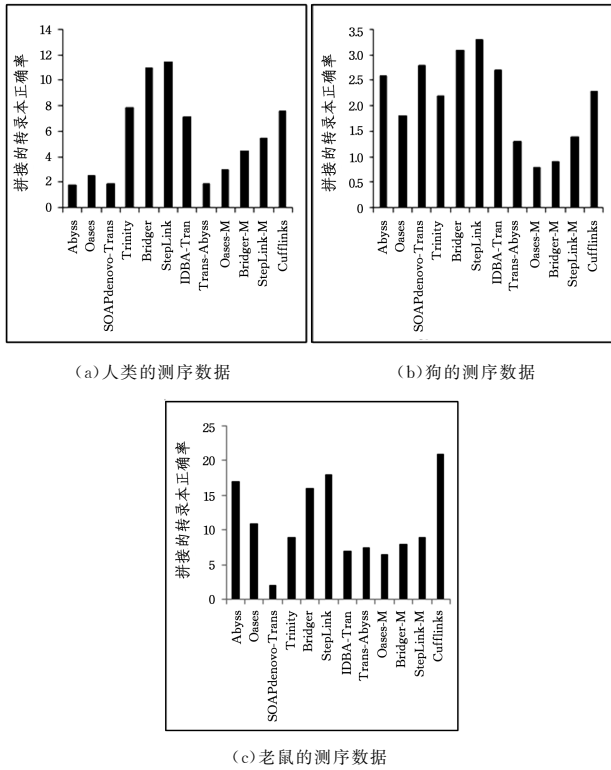


图 7 对 3 种 RNA 测序数据采用不同算法所得的精确度对比
Fig. 7 Accuracy comparison of different algorithms for three RNA sequencing datasets

4.3 算法复杂度分析

在时间方面,StepLink 的运算速度比其他所有算法都快。本研究使用了相同的平台,测试 CPU 为 Intel 至强处理器 (2.4GHz)。对于不同物种,6 个单 k 值算法的比对结果如图 8 所示。特别说明,由于 Cufflinks 是基于参考基因组的转录组拼接算法,在拼接时需要用到映射软件 TopHat,其运行时间较长,与从头拼接算法所用时间没有可比性,因此本文不对其进行对比;各算法的 M 值、Trans-Abyss 和 IDBA-Tran 都使用了多个 k 值,需要不断运行程序以达到最好的效果,这个过程会消耗大量时间,与图 8 中的 6 个单 k 值算法所用时间没有可比性,因此本文也不对其进行对比。从对比情况可知,StepLink 在运行速度上具有明显的优势。相比于 Trinity,StepLink 的运行时间大幅缩短。即使与最新的 Bridger 相比,StepLink 的效率也有所提升。

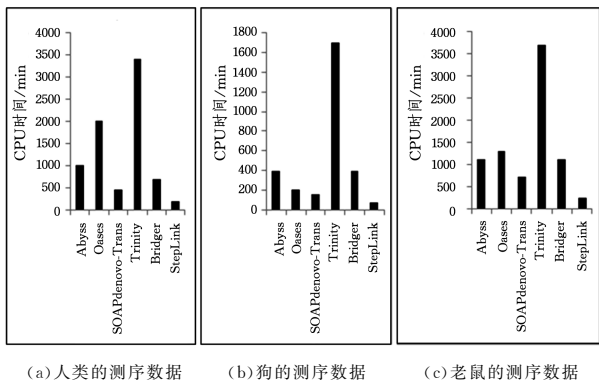


图 8 不同算法使用 3 种 RNA 测序数据的 CPU 时间对比
Fig. 8 CPU time comparison of different algorithms using three RNA sequencing datasets

结束语 本研究针对转录组拼接提出了一种新的转录组从头拼接算法 StepLink,该算法不仅在敏感度方面优于其他算法,而且保证了很高的准确度,同时提升了运行速度。StepLink 针对不同物种的数据均有很好的鲁棒性,尤其是针对低表达的基因数据。本文的研究思路可为转录组的研究提供帮助,也为项目组下一步对基于参考基因组的转录组拼接算法的研究奠定了基础。

参考文献

- [1] YU A M. Research on the sugar and terpenoid metabolism during the Amomum villosum Lour. fruit development using RNA-Seq [D]. Guangzhou: Guangzhou University of Chinese Medicine, 2014. (in Chinese)
于安民. 基于 RNA-Seq 的阳春砂果实发育过程中糖和萜类代谢的研究[D]. 广州: 广州中医药大学, 2014.
- [2] QI Y X, LIU Y B, RONG W H. RNA-Seq and its applications; a new technology for transcriptome [J]. Hereditas, 2011, 33 (11): 1191-1202. (in Chinese)
祁云霞, 刘永斌, 荣威恒. 转录组研究新技术: RNA-Seq 及其应用[J]. 遗传, 2011, 33(11): 1191-1202.
- [3] LU Z Y. Research on assembly algorithm for next generation sequencing technology [D]. Nanjing: Southeast University, 2011. (in Chinese)
卢志远. 面向新一代测序技术的拼接算法研究[D]. 南京: 东南大学, 2011.
- [4] PERTEA G. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation [J]. Nature Biotechnology, 2010, 28(5): 511-515.
- [5] MINGFU S, CARL K. Accurate assembly of transcripts through phase-preserving graph decomposition [J]. Nature Biotechnology, 2017, 35(12): 1167-1169.
- [6] LIU J T, YU T, JIANG T, et al. TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs [J]. Genome Biology, 2016, 17(1): 213.
- [7] PERTEA M, PERTEA GM, ANTONESCU C M, et al. StringTie enables improved reconstruction of a transcriptome from RNA-Seq reads [J]. Nature Biotechnology, 2015, 33(3): 290-295.
- [8] MARETTY L, SIBBESEN J A, KROGH A, et al. Bayesian transcriptome assembly [J]. Genome Biology, 2014, 15 (10): 501.
- [9] SCHULZ M H, ZERBINO D R, Vingron M, et al. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels [J]. Bioinformatics, 2012, 28(8): 1086-1092.
- [10] XIE Y, WU G, TANG J, et al. SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads [J]. Bioinformatics, 2014, 30(12): 1660.
- [11] PENG Y, LEUNG H C, YIU S M, et al. IDBA-Tran: a more robust de novo de Bruijn graph assembler for transcriptomes with uneven expression levels [J]. Bioinformatics, 2013, 29(13): 326-334.
- [12] GRABHERR M G, HAAS B J, YASSOUR M, et al. Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data [J]. Nature Biotechnology, 2011, 29 (7): 644-652.
- [13] CHANG Z. De novo transcriptome assembly from RNA-Seq

- [D]. Jinan; Shandong University, 2014. (in Chinese)
常征. 基于RNA测序技术的转录组从头拼接算法研究[D]. 济南: 山东大学, 2014.
- [14] ZHENG C, LI G, LIU J, et al. Bridger: a new framework for de novo transcriptome assembly using RNA-seq data [J]. *Genome Biology*, 2015, 16(1): 30.
- [15] XIONG X J. Introduction to NCBI's SRA database [J]. *Chemistry of Life*, 2010(6): 959-963. (in Chinese)
熊筱晶. NCBI高通量测序数据库SRA介绍[J]. *生命的化学*, 2010(6): 959-963.
-
- (上接第298页)
- [6] WANG L J, CHEN Y, YAN X C, et al. Differential Data Update Scheme on Network-Coding-Based Cloud Storage System [J]. *Journal on Communications*, 2017, 38(3): 154-164. (in Chinese)
王龙江, 陈越, 严新成, 等. 网络编码云存储系统差分数据更新方案[J]. *通信学报*, 2017, 38(3): 154-164.
- [7] WU H, LAI C Z, FAN J L, et al. Data Update Algorithm Based on Secure Network Coding in Cloud Environment [J]. *Journal on Communications*, 2017, 35(5): 121-127. (in Chinese)
吴昊, 赖成喆, 范九伦, 等. 云环境下基于安全网络编码的数据更新方法[J]. *通信学报*, 2017, 35(5): 121-127.
- [8] WU Y, DIMAKIS A G. Reducing Repair Traffic for Erasure Coding-Based Storage via Interference Alignment [C] // 2009 IEEE International Symposium on Information Theory. Seoul: IEEE Press, 2009: 2276-2280.
- [9] CADAMBE V R, JAFAR S A. Interference Alignment and the Degrees of Freedom for the K User Interference Channel [J]. *IEEE Transactions on Information Theory*, 2008, 54(8): 3425-3441.
- [10] HUANG Q. Efficient Repair Schemes in Cloud Storage Based on Interference Alignment [D]. Chongqing: Chongqing University of Posts and Telecommunications, 2014. (in Chinese)
黄倩. 基于干扰对齐的高效云存储修复方法研究[D]. 重庆: 重庆邮电大学, 2014.
- [11] CHEN H C H, HU Y, LEE P P C, et al. NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds [J]. *IEEE Transactions on Computers*, 2014, 63(1): 31-44.
- [12] ZHANG J F. Cloud Storage Schemes Based on Network Coding [D]. Changsha: Central South University, 2014. (in Chinese)
张俊峰. 基于网络编码云存储方案研究[D]. 长沙: 中南大学, 2014.
- [13] XIE C Y, JIA Z T, QING S H, et al. Semi-Random Linear Network Coding for Cloud Storage Redundancy [J]. *Journal of Beijing University of Posts and Telecommunications*, 2013, 36(3): 30-34. (in Chinese)
谢垂益, 贾忠田, 卿斯汉, 等. 适用于云存储冗余的半随机线性网络编码[J]. *北京邮电大学学报*, 2013, 36(3): 30-34.
- [14] XU Y, HE Q, LUO Y. Optimal Repair for Distributed Storage Codes in Vehicular Networks [C] // 2016 IEEE 83rd Vehicular Technology Conference. Nanjing: IEEE Press, 2016: 1-5.
- [15] SIPOS M, GAHM J, VENKAT N, et al. Erasure Coded Storage on A Changing Network: the Untold Story [C] // 2016 IEEE Global Communications Conference (GLOBECOM). Washington DC: IEEE Press, 2016: 1-6.
- [16] ELYASI M, MOHAJER S. Determinant Coding: A Novel Framework for Exact-Repair Regenerating Codes [J]. *IEEE Transactions on Information Theory*, 2016, 62(12): 6683-6697.
- [17] WANG Z, TAMO I, BRUCK J. Explicit Minimum Storage Regenerating Codes [J]. *IEEE Transactions on Information Theory*, 2016, 62(8): 4466-4479.
- [18] PAPALIOPOULOS D S, LUO J, DIMAKIS A G, et al. Simple Regenerating Codes: Network Coding for Cloud Storage [OL]. <http://arXiv.org/pdf/1109.0264.pdf>.
- [19] PRAKASH N, KRISHNAN M N. The Storage-Repair-Bandwidth Trade off of Exact Repair Linear Regenerating Codes for the Case $d=k=n-1$ [C] // IEEE International Symposium on Information Theory (ISIT). Hong Kong: IEEE Press, 2015: 859-863.
- [20] LI J, TANG X, XIANG W. A New Construction of $(k+2, k)$ Minimal Storage Regenerating Code over F_3 with Optimal Access Property for All Nodes [J]. *IEEE Communications Letters*, 2016, 20(7): 1289-1292.
- [21] WANG Z, TAMO I, BRUCK J. Optimal Rebuilding of Multiple Erasures in MDS Codes [J]. *IEEE Transactions on Information Theory*, 2017, 63(2): 1084-1101.
- [22] LI R, LIN J, LEE P P C. Enabling Concurrent Failure Recovery for Regenerating-Coding-Based Storage Systems: from Theory to Practice [J]. *IEEE Transactions on Computers*, 2015, 64(7): 1898-1911.
- [23] ZHANG H, LI H, SHUM K W, et al. Concurrent Regenerating Codes [J]. *IET Communications*, 2017, 11(3): 362-369.
- [24] KUMAR S, ROSNES E, AMAT A G I. Secure Repairable Fountain Codes [J]. *IEEE Communications Letters*, 2016, 20(8): 1491-1494.
- [25] AGARWAL A, MAZUMDAR A. Security in Locally Repairable Storage [J]. *IEEE Transactions on Information Theory*, 2016, 62(11): 6204-6217.
- [26] CHEN Y, WANG L, LIAO C. Eavesdropping Prevention for Network Coding Encrypted Cloud Storage Systems [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2016, 27(8): 2261-2273.