基因组重组排序算法综述

崔 筠 朱大铭 马绍汉
(山东大学计算机科学与技术学院 济南 250061)

摘 要 随着快速测序技术的发展，对大规模DNA分子的研究与其中的基因相对次序有关。基因组重组是计算生物学的一个重要研究领域，是基因组在基因水平比较分析的基础。其研究目标是找最短的重组操作序列，将一种基因组转变为另一种基因组。基于分子生物学的实验证明，这种序列有助于估计不同基因组间的进化事件。基因组进化过程虽然非常复杂，但可用3种基本的重组操作模拟，即反转(reversal)、移位(translocation)和转位(transposition)。本文讨论了这些操作相关的重组算法以及各种排序距离的计算方法。

关键词 基因组重组，排序距离，反转，移位，转位

A Survey of Sorting Algorithms for Genome Rearrangements

CUI Yun ZHU Da-Ming MA Shao-Han
(School of Computer Science and Technology, Shandong University, Ji'nan 250061)

Abstract With the development of fast sequencing techniques, large-scale DNA molecules are investigated with respect to the relative order of genes in them. Genome rearrangement is an important area of computational biology, and bases the comparison analysis of genomes at the level of genes. The goal is to find the shortest sequence of genome arrangements operations that transform one genome into another. Such sequence is helpful to estimate the evolutionary events between different genomes, which is proved by many tests of molecule biology. Although the evolutionary process between genomes is very complicated, there are three basic rearrangement operations: reversal, translocation, transposition. In this paper, we discuss the rearrangement algorithms for these operations, and the methods to compute various types of sorting distances.

Keywords Genome rearrangement, Sorting distance, Reversal, Translocation, Transposition

1 引言

随着人类基因组计划的实施，生物学的数据信息飞速增长。如何从这些海量数据中提取有用的知识，揭示这些数据所蕴含的生物学意义，是对计算机科学的巨大挑战。由此引出了一门新兴学科：计算生物学。计算生物学涵盖了生物学、数学和计算机科学的综合应用，已经成为当今生命科学和自然科学的核心领域和最具活力的前沿领域之一。

基因组重组是计算生物学的一个重要研究领域，是基因组在基因水平比较分析的基础。上世纪80年代末，Jeffrey Palmer等人比较了卷心菜和芜菁的基因组，发现它们包含大量相似的基因(许多基因的相似度达99%～99.9%)，区别在于这些基因在染色体上的排列次序不同。1984年，Nadean和Taylor估计人与老鼠的基因组间只相差178±39次的重组操作。随后，Copeland等人在1993年根据新绘的人和老鼠基因组联接图，验证了该估计。随后的许多研究表明，基因组重组是生物进化的一种普遍模式，也是植物、哺乳动物及细菌等呈现多样性的主要原因之一。

基因组进化过程虽然非常复杂，但可用3种主要操作模拟，即反转(reversal)、移位(translocation)和转位(transposition)。反转和转位操作是在一条染色体上重组基因序列片断，而移位操作是在两条染色体间交换基因片断。一次重组操作将一个基因组转化为另一个新的基因组。

给定两个基因组，重组排序问题是要计算一个重组操作序列，将其中一个基因组转化为另一个，并使得重组操作次数最少。基于分子生物学积累的实验数据验证，重组操作次数最少的基因组重组操作序列可以较好地估计基因组间的进化关系，从而推断物种间的实际进化过程。

2 重组操作与排序距离

基因组是一组染色体的集合，一条染色体是一个基因序列，表示为一个整数序列。有符号基因组中，每个基因均采用带符号的整数表示，而无符号基因组中的基因则用正整数表示。

包含N条染色体的基因组A表示为$A=\{X(1),X(2),\cdots,X(N)\}$，其中$X(i)=x_1(i),x_2(i),\cdots,x_{n_i}(i)$。一个基因组中任意两个基因均不相同，$x_1(i)$和$x_{n_i}(i)$称为染色体$X(i)$的头基因，其它基因称为$X(i)$的中间基因。设$X,Y$为两条有向染色体，若$X=Y$或$X=-Y$，则$X$与$Y$实际表示一条染色体，称$X$与$Y$等价。若基因组$A$，$B$含有相同的染色体集合，则称基因组$A$，$B$等价。

假定重组排序操作$\rho(X,Y,i,j)$将基因组A转换为另一基因组A_1，记为$A\cdot\rho(X,Y,i,j)=A_1$。

定义2.1 给定基因组A，B，重组排序问题是计算由A

崔 筠 博士生，主要研究计算生物学；朱大铭 教授，博士生导师，主要研究计算生物学、算法分析与设计；马绍汉 教授，博士生导师，主要研究计算生物学、算法分析与设计。

转换为 B 的排序操作序列 $\rho_1,\rho_2,\cdots,\rho_k$，即 $A\cdot\rho_1\cdot\rho_2\cdot\cdots\cdot\rho_k=B$，使排序操作次数 k 最小。k 的最小值称为 A,B 的排序距离，记为 $d(A,B)$。

排序距离对基因组间的进化关系，提供了一个很好的估计值，有助于确定物种间的亲缘关系。根据重组操作的不同，排序距离有多种类型，如反转距离、移位距离及转位距离等。

定义 2.2 反转(Reversal)操作 $\rho(i,j)$ 作用在一条染色体 $X=x_1,\cdots,x_{i-1},x_i,x_{i+1},\cdots,x_{j-1},x_j,x_{j+1},\cdots,x_n$ 上，其结果为一条新染色体 $X'=x_1,\cdots,x_j,x_{j-1},\cdots,-x_{i+1},x_i,\cdots,x_n$。

反转操作是将一个基因序列的基因次序颠倒。如果基因有符号，则把符号取反。给定一条染色体 X，把 X 转换为另一条染色体 Y 所需的最少反转操作次数，称为染色体 X 和 Y 的反转距离。

定义 2.3 给定两条染色体 $X=x_1,x_2,\cdots,x_m$ 和 $Y=y_1,y_2,\cdots,y_n$，假设作用在 X 和 Y 上的移位(Translocation)操作把 X 和 Y 分别断裂为：$X=X_1,X_2$，$Y=Y_1,Y_2$，$X_1=x_1,\cdots,x_{i-1}$，$X_2=x_i,\cdots,x_m$，$Y_1=y_1,\cdots,y_{j-1}$，$Y_2=y_j,\cdots,y_n$。前前移位操作 $\rho_{pp}(X,Y,i,j)$ 产生两条新的染色体 $X_1Y_2=x_1,\cdots,x_{i-1},y_j,\cdots,y_n$ 和 $Y_1X_2=y_1,\cdots,y_{j-1},x_i,\cdots,x_m$。而前后移位操作 $\rho_{ph}(X,Y,i,j)$ 也产生两条新的染色体，如果基因是无符号的，则为 $X_1Y_1=x_1,\cdots,x_{i-1},y_{j-1},\cdots,y_1$ 和 $X_2Y_2=x_m,\cdots,x_i,y_j,\cdots,y_n$。如果基因是有符号的，则为 $X_1-Y_1=x_1,\cdots,x_{i-1},-y_{j-1},\cdots,-y_1$ 和 $-X_2Y_2=-x_m,\cdots,-x_i,y_j,\cdots,y_n$。

移位操作是将两条染色体分别断开，再重新连接成两条新的染色体。前前移位不改变基因的符号，而前后移位可能会将基因的符号取反。对包含多条染色体的基因组，把它转换为目标基因组所需的最少移位操作次数，称为这两个基因组间的移位距离。

定义 2.4 转位(Transposition)操作 $\rho(i,j,k)$ 作用在一条染色体 $X=x_1,\cdots,x_{i-1},x_i,\cdots,x_{j-1},x_j,\cdots,x_{k-1},x_k,\cdots,x_n$ 上，其结果为一条新染色体 $X'=x_1,\cdots,x_{i-1},x_j,\cdots,x_{k-1},x_j,\cdots,x_{j-1},x_k,\cdots,x_n$。

转位操作是将一条染色体上的两个基因序列交换位置，并且不会改变基因的符号。给定一条染色体 X，把 X 转换为另一条染色体 Y 所需的最少转位操作次数，称为染色体 X 和 Y 的转位距离。

3 断点图

Bafna，Pevzner 首次引入断点图，揭示了断点图的最大圈分解与基因组的反转排序间存在密切的联系。之后，几乎所有关于基因组重组排序的算法研究结果均以断点图作为工具。

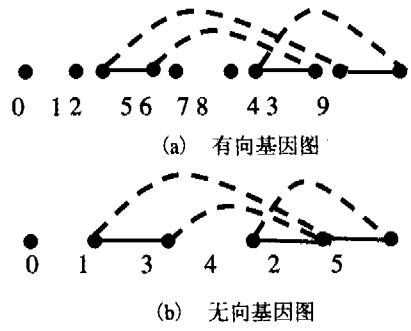
定义 3.1 定义 $i\sim j$ 为 $|i-j|=1$。在一个基因序列 $\pi_1\pi_2\cdots\pi_n$ 前后加入 $\pi_0=0$ 和 $\pi_{n+1}=n+1$。对 π 中的一对元素 (π_i,π_{i+1})，$0\leqslant i\leqslant n$，如果 $\pi_i\sim\pi_{i+1}$，则称为邻接；否则，称为断点。

定义 3.2 基因序列 $\pi_1\pi_2\cdots\pi_n$ 对应的断点图有 $n+2$ 个顶点 $\{\pi_0,\pi_1,\cdots,\pi_n,\pi_{n+1}\}=\{0,1,\cdots,n,n+1\}$。若 (π_i,π_j) 在 π 中是断点($\pi_i!\sim\pi_j$ 且 $i\sim j$)，则用黑边连接 π_i 和 π_j；若 (i,j) 在 π^{-1} 中是断点($\pi_i\sim\pi_j$ 且 $i!\sim j$)，则用灰边连接 π_i 和 π_j。

对于一个基因序列，断点图的顶点既和基因相关，也和基因的符号相关。无符号基因序列中的每个基因在断点图中用一个顶点表示。而有符号基因序列中的每个基因在断点图中用一对相邻的顶点表示，即把序列中的正元素 $+x$ 用 $2x-1$，$2x$ 代替，负元素 $-x$ 用 $2x$，$2x-1$ 代替，把有符号序列等价转换为无符号序列。

例 1 设有符号基因序列 $\pi=1,3,4,-2$，则其断点图如图 1(a)所示。

例 2 设无符号基因序列 $\pi=1,3,4,2$，则其断点图如图 1(b)所示。



(a) 有向基因图

(b) 无向基因图

图 1 (a)有向基因图；(b)无向基因图

在有符号基因组的断点图 $Gs(A,B)$ 中，每个顶点的或者是孤立的，或者只和一条黑边和一条灰边相连，因此断点图 $Gs(A,B)$ 可以唯一分解为黑边和灰边交替的圈。圈数用 c 表示。

在无符号基因组的断点图 $G(A,B)$ 中，每个顶点或者为孤立点，或者和一条黑边及一条灰边相连，或者和两条黑边及两条灰边相连。$G(A,B)$ 可以有多种圈分解。每种分解都可看成是对所有基因进行符号指派。一旦我们求得断点图 $G(A,B)$ 的一个圈分解，实际是给每个基因指定了一个符号。

虽然有符号基因组的 Reversal 和 Translocation 排序已有一些多项式时间算法，其中最好的时间复杂度为 $O(n^2)$(n 为基因个数)，但是许多生物基因组数据并未给出基因方向，因此研究无符号基因组的排序算法亦具有十分重要的价值。求无符号基因组排序距离的一个方法是：(1)尝试各种可能的圈分解(为每个基因指定符号)；(2)计算各个圈分解对应的有符号基因组的排序距离，选取其中的最小值。

一般假定目标基因组为递增的整数排列，不存在断点。因此，基因组的重组操作也就是如何消除断点的过程。已经证明，基因组的排序距离与断点图的圈分解及断点图中特定的组合结构密切相关。

4 反转排序

大约 70 年前，Dobzhansky 和 Sturtevant 在研究果蝇基因组时发现，有一种果蝇的基因组，可以通过 17 次反转操作变为另一种果蝇的基因组。随后的研究表明，反转是基因组重组中最常见的事件，反转距离则对判断两个物种间的进化关系提供了一个较好的估计值。基于数学理论分析的方法研究基因组重组是由 Sankoff 等人开始的。根据研究的基因组是否有符号，近年来出现的反转排序的相关结果，可以分为有符号反转排序和无符号反转排序两种类型。

4.1 有符号基因组反转排序

对有符号基因组反转排序问题，Bafna 和 Pevzner 首先给出了近似度为 1.5 的多项式算法[1]，并同时引入断点图的概念，揭示了图的最大环分解和反转距离之间的联系。Hannenhalli 和 Pevzner 利用断点图解决了有符号基因组反转排序问题，设计出 $O(n^5)$ 多项式算法[2]；随后，基于断点图的一

些特定的组合结构，Hannenhalli 和 Pevzner 又将该算法的时间复杂度改进为 $O(n^2\alpha(n))$，此处 α 是阿克曼函数的倒数。Kaplan 与 Tarjian 设计了一个更简单也更快的算法，时间复杂度为 $O(n^2)$[3]。最近，Bader 等给出一个在 $O(n)$时间内计算反转距离的算法[4]，也给出了在 $O(n^2)$时间内计算反转排序序列的算法。

有符号反转距离 $d_s(A,B)$的下界最早是用断点图中的黑边数目 b 和圈数 c 表达的，即 $d_s(A,B)\geq b-c$。Hannenhalli 和 Pevzner 揭示了一个称为 hurdle 的隐藏参数，其数目一般用 h 表示，从而证明了 $d_s(A,B)$的上界为 $n-N-c+h+1$，下界为 $n-N-c+h$。基于这些，他们在多项式时间内解决了有符号反转排序问题。引理 4.1.1 给出了计算反转距离的公式，此处 $f=0$ 或 1。

引理 4.1.1(Hannenhalli)　给定有符号基因组 A 和 B，

$d_s(A,B)=n-N-c+h+f$

4.2　无符号基因组反转排序

Kececioglu 和 Sankoff 首先对无符号基因组反转排序进行研究，给出了一个近似度为 2 的贪心算法[5]，其时间复杂度为 $O(n^2)$，所需空间为 $O(n)$。同时也给出了一个剪枝限界的精确算法，在 $O(m\cdot L(n,n))$时间内找到最优解，空间复杂度为 $O(n^2)$。此处 m 是剪枝限界搜索树的大小，$L(n,n)$表示求解包含 n 个变量和 n 个常量的线性规划所需的时间。Hannenhalli 和 Pevzner 证明[6]，对任意基因序列，存在一个最优的反转序列，其每次操作都不会增加断点。揭示了 singleton 结构(一个基因，其左边和右边都构成断点)是求解反转排序问题的难点，并且对不包含 singleton 结构的无符号基因组，给出了反转排序的多项式算法。

Capara 于 1997 年证明无符号基因组反转排序问题为 NP-hard[7]。他先把一个称为最大欧拉圈分解的 NP-hard 问题归约为断点图的最大圈分解问题，然后再把断点图最大圈分解归约为反转排序问题。Berman 和 Karpinski 证明了无符号基因组反转排序问题是 MAX SNP-hard 的[8]，其多项式近似算法的近似度下界为 1.0008，也即除非 P=NP，该问题不可能存在优于 1.0008 倍的多项式近似算法。

Bafna 和 Pevzner 设计了无符号基因组反转排序的 1.75 倍近似算法，时间复杂度为 $O(n^2)$。1998 年，Christie 给出了该问题的近似度为 1.5 的多项式近似算法[9]，时间复杂度为 $O(n^4)$。Berman 等应用有符号反转排序的多项式算法及新的圈分解近似算法，设计了该问题的近似度为 1.375 的多项式近似算法[10]。

5　移位排序

移位重组在哺乳动物进化中很常见，与反转操作存在许多十分类似的性质，但由于移位涉及一个基因组的两条染色体，因此移位排序显得更为复杂。移位排序一般假定两个基因组包含相同数目的染色体，具有相同的头基因集合。一个作用在染色体 $X=x_1,x_2,\cdots,x_m$ 和 $Y=y_1,y_2,\cdots,y_n$ 上的移位排序 $\rho(X,Y,i,j)$可以看成是作用在 $X-Y$ 上的一个反转操作 $\rho(X-Y,i,m+(n-j+1))$，但该反转不一定是最优的。

5.1　有符号基因组移位排序

Hannenhalli 首次设计出有符号基因组移位排序的 $O(n^3)$多项式算法[11]，适用于前前移位和前后移位两种模式。Zhu 等将计算移位距离的时间复杂度由 $O(n^3)$改进为 $O(n^2)$，将计算移位序列的时间复杂度由 $O(n^3)$改进为 $O(n^2\log n)$[12]。Wang 等进一步将计算移位序列的时间复杂度改进为 $O(n^2)$[13]。Li 等给出了在 O(n)时间内计算移位距离的算法[14]。

有符号基因组的移位距离和圈数及最小子排列(断点图中的一种组合结构)的数目紧密相关。设两个基因组 A 和 B 分别包含 n 个基因和 N 条染色体。断点图 $G_s(A,B)$中的圈数和最小子排列数目分别设为 c 和 s。参数 $f=0$ 或 1 或 2。引理 5.1.1 给出了计算移位距离的公式。

引理 5.1.1(Hannenhalli)　给定有符号基因组 A 和 B，

$d_s(A,B)=n-N-c+s+f$

5.2　无符号基因组移位排序

Kececioglu 和 Ravi 将无符号基因组移位问题转化为交换字符串的前缀和后缀，给出了一个近似度为 2 的贪心算法[15]。最近，Zhu 和 Wang 证明了无符号基因组移位排序问题是 NP-Hard 的[16]，其多项式近似算法的近似度下界为 1.00017。Cui 等应用最大匹配理论及断点图圈分解的一些新性质[17]，给出了近似度为 1.75 的无符号基因组移位排序算法，运行时间为 $O(n^2)$。

6　转位排序

转位排序与反转排序的相似之处在于，都是作用在一条染色体上的操作。但转位排序目前的复杂性尚不清楚，是否是多项式可解的或者是 NP-Hard 问题还没有定论。Bafana 与 Pevzner 首先给出了近似度为 1.5 的多项式近似算法[18]，运行时间是 $O(n^2)$，并确定转位排序的上界为 $n+1-c$，下界为 $\frac{1}{2}(n+1-c_{odd})$。此处 c 和 c_{odd}分别表示断点图中的圈数及包含黑边数目为奇数的圈数。Christie 给出了另外一个比较简单的近似度为 1.5 的近似算法[19]，运行时间是 $O(n^4)$。Hartman 证明了环形排列和线性排列的转位排序问题是等价的[20]，从而设计了近似度为 1.5 的近似算法，运行时间为 $O(n^{3/2}\sqrt{\log n})$。最近，基于计算机辅助程序的实例分析验证，Isaac 和 Hartman 设计了近似度为 1.375 的多项式近似算法[21]，其时间复杂度为 $O(n^2)$。

7　多种操作排序

上面讨论的都是单一的基因组重组操作。实际的生物进化过程相当复杂，可能涉及到多种重组操作。下面我们将讨论两种典型的多种操作同时存在的重组排序。

7.1　反转与转位排序

Walter 等人对有符号基因组的反转与转位排序给出了近似度为 2 的近似算法[22]，对无符号基因组的反转与转位排序，给出了近似度为 3 的近似算法。另外，由转位操作衍生了一种称为反转位的操作，即某个基因序列片段先进行反转，再转位到同一条染色体上的其它位置。见定义 7.1.1。Gu 等人对有符号基因组的反转与反转位排序给出了近似度为 2 的近似算法[23]。Hartman 和 Sharan 设计了近似度为 1.5 的近似算法[24]，其时间复杂度为 $O(n^{3/2}\sqrt{\log n})$。

定义 7.1.1　反转位操作 $\rho(i,j,k)$作用在一条染色体 $X=x_1,\cdots,x_{i-1},x_i,\cdots,x_{j-1},x_j,\cdots,x_{k-1},x_k,\cdots,x_n$ 上，其结果为一条新染色体 $X'=x_1,\cdots,x_{i-1},x_j,\cdots,x_{k-1},x_{j-1},\cdots,x_j,x_k,\cdots,x_n$。

7.2　反转与移位排序

Kececioglu 与 Ravi 首先对基因组反转和移位操作并存的问题进行研究，对有符号基因组给出了近似度为 1.5 的多项式时间算法，同时对无符号基因组给出了近似度为 2 的多项式时间算法，这两个算法的时间复杂度都是 $O(n^2)$。Hannenhalli 和 Pevzner 设计了有符号基因组反转和移位操作并存的的多项式算法[25]，并给出了排序距离的计算公式。

总结 随着快速测序技术的发展，对大规模 DNA 分子的研究与其中的基因相对次序有关。基因组重组排序为重构生物进化的相对关系提供了较好的估计值，并已经被大量的生物学实验数据所验证。本文讨论了几种典型的基因组重组操作及其研究进展。

参 考 文 献

1 Bafna V, Pevzner P. Genome rearrangements and sorting by reversals. FOCS, 1993. 148～157
2 Hannenhalli S, Pevzner P A. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). STOC, 1995. 178～189
3 Kaplan H, Shamir R, Tarjan R E. Faster and simpler algorithm for sorting signed permutations by reversals. SIAM Journal of Computing, 2000, 29(3): 880～892
4 Bader D A, Moret B M E, Yan Mi. A Linear-Time Algorithm for Computing Inversion Distance Between Signed Permutations with an Experimental Study. Journal of Computational Biology, 2001, 8(5): 483～491
5 Kececioglu J, Sankoff D. Exact and approximation algorithms for the inversion distance between two permutations. CPM, 1993. 87～105
6 Hannenhalli S, Pevzner P A. To cut … or not to cut (applications of comparative physical maps in molecular evolution). SODA, 1996. 304～313
7 Caprara A. Sorting by reversals is difficult. RECOMB, 1997. 75～83
8 Berman P, Karpinski M. On some tighter inapproximability results. Electronic Colloquium on Computational Complexity (ECCC), 1998, 5(29)
9 Christie D A. A 3/2 Approximation Algorithm for Sorting by Reversals. SODA, 1998. 244～252
10 Berman B, Hannenhalli S, Karpinski M. 1.375-approximation algorithm for sorting by reversals. ESA, 2002. 200～210
11 Hannenhalli S. Polynomial-time Algorithm for Computing Translocation Distance Between Genomes. CPM, 1995. 162～176
12 Zhu Daming, Ma Shaohan. Improved polynomial-time algorithm for computing translocation distance between genomes. The Chinese Journal of Computers (in Chinese), 2002, 25(2): 189～196
13 Wang Lusheng, Zhu Daming, Liu Xiaowen, An $O(n^2)$ algorithm for signed translocation. APBC, 2005. 349～358
14 Li G, Qi X, Wang X, et al. A linear-time algorithm for computing translocation distance between signed genomes. CPM, 2004
15 Kececioglu J, Ravi R. Of mice and men: Algorithms for evolutionary distances between genomes with translocation. SODA, 1995. 604～613
16 Zhu Daming, Ma Shaohan, Wang Lusheng. Sorting unsigned genome by translocation is NP-hard. Theor Comput Sci, 2006, 352(1-3): 322～328
17 Cui Yun, Wang Lusheng, Zhu Daming. A 1.75-Approximation Algorithm for Unsigned Translocation Distance. ISAAC, 2005. 392～401
18 Bafna V, Pevzner P A. Sorting by transpositions. SIAM Journal on Discrete Mathematics, 1998, 11(2): 224～240
19 Christie D A, Irving R W. Sorting Strings by Reversals and by Transpositions. SIAM J Discrete Math, 2001, 14(2): 193～206
20 Hartman T. A Simpler 1.5-Approximation Algorithm for Sorting by Transpositions. CPM, 2003. 156～169
21 Elias I, Hartman T. A 1.375-Approximation Algorithm for Sorting by Transpositions. WABI, 2005
22 Walter M E T, Dias Z, Meidanis J. Reversal and Transposition Distance of Linear Chromosomes. SPIRE, 1998. 96～102
23 Gu Qian-Ping, Peng Shietung, Sudborough I H. A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions. Theor Comput Sci, 1999, 210(2): 327～339
24 Hartman T, Sharan R. A 1.5-approximation algorithm for sorting by transpositions and transreversals. J Comput Syst Sci, 2005, 70(3): 300～320
25 Hanennhalli S, Pevzner P A. Transforming Men into Mice (Polynomial Algorithm for Genomic Distance Problem). FOCS, 1995. 581～592

（上接第 82 页）

5.4 删除网络封包记录

删除网络封包记录函数 DeleteSession 定义为 int DeleteSession(SOCKET s)，输入的参数 s 为 Socket 标志，用来区分不同的网络封包。它的主要功能是从已有的网络封包记录中删除一条，在删除之前要做的是调用 SendSessionToApp 函数，将删除的记录发给执行文件。

5.5 寻找功能

寻找函数 FindSession 定义为 int FindSession (SOCKET s)。输入的参数 s 为要查找的的网络封包记录的标志符。函数返回的是网络封包结构在数组中的索引，但如果返回的索引值大于网络数据封包总数，则表示没有找到。

5.6 设置功能

设置函数 SetSession 的定义为 int SetSession(SESSION * session, BYTE bDirection, UINT uiPort, DWORD ulRemoteIP)。输入和输出的参数 session 为指向要设置的网络封包数据结构，输入参数 bDirection 为连接的进出方向，输入参数 uiPort 为连接的目的端口，输入参数 ulRemoteIP 为连接的目的 IP 地址。它的主要功能是对一个已经存在的网络封包记录进行修改，修改的字段有：协议类型、进出方向、目的端口和目的 IP 地址。

5.7 修改记录动作

修改记录动作函数 SetSessionEx 定义为 int SetSessionEx (SESSION * session, BYTE bDirection, const TCHAR * pMemo, int ByteCount, BOOL isSend)。它与 SetSession 的不同是，其修改字段为：本地端口/IP、进出方向、备注信息及进出流量。输入参数 pMemo 是连接的备注信息，输入参数 isSend 标识是发送还是接收的流量，TRUE：ByteCount 表示出流量；FALSE：ByteCount 表示进流量。

结论 数据包的解析技术是从截获的数据包中收集比较重要的信息，定义了一个封包结构体来存储这些关键的信息，然后在利用该封包存储的信息去判别类似的数据包的行为，实现防火墙控制数据流通的功能：数据包的监视，实现可视化的数据包监视界面，提供清空监视列表，停止/开始监视及停止/开始滚动功能；控管规则的设置，包括自定义添加，修改及删除控管规则，设置网络 IP 地址段，设置访问时间等；系统参数配置，是否自动启动，是否声音报警，是否闪烁图标报警等。

参 考 文 献

1 Firewall is security device of choice. Australian Electronics Engineering, 2002, 000(M4): E123～E123
2 Wasti S. Hardware assisted packet filtering firewall. In: Proceedings of the 2000-2001 Grad Symposium, CS Dept, University of Saskatchewan, April 2001
3 Yamagaki N, Minami K. Packet Discarding Scheme Considering Both Instantaneous and Historical Use of Network Resources. IEICE Transactions on Communications, 2001, E84-B(8): 2115～2123
4 Li Xin. Stateful Inspection Firewall Session Table Roces Sing [J]. International, 2005, (2): 615～620
5 Law K L E, Leung R. A design and implementation of active network socket programming. Microprocessors and Microsystems, 2003, 27(5-6): 277～284