Genomes containing Duplicates are Hard to compare (Extended abstract)

Cedric Chauve
LaCIM, CGL, Département d'Informatique, Université du Québec À Montréal

Guillaume Fertin
LINA FRE CNRS 2729, Université de Nantes

Romeo Rizzi
Dipartimento di Matematica e Informatica - Università di Udine

Stéphane Vialette
LRI, UMR CNRS 8623, Université Paris-Sud

In *2nd International Workshop on Bioinformatics Research and Applications, IWBRA 2006*, volume 3992 of *Lecture Notes in Computer Science*, pages 783–790.

Genomes containing Duplicates are Hard to compare (Extended Abstract)*

Cedric Chauve[1], Guillaume Fertin[2], Romeo Rizzi[3], and Stéphane Vialette[4]

[1] LaCIM, CGL, Département d'Informatique, Université du Québec À Montréal
CP 8888, Succ. Centre-Ville, H3C 3P8, Montréal (QC) - Canada
chauve@lacim.uqam.ca
[2] Laboratoire d'Informatique de Nantes-Atlantique (LINA), FRE CNRS 2729
Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3 - France
fertin@lina.univ-nantes.fr
[3] Dipartimento di Matematica e Informatica - Università di Udine - Italy
Romeo.Rizzi@dimi.uniud.it
[4] Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623
Faculté des Sciences d'Orsay - Université Paris-Sud, 91405 Orsay - France
vialette@lri.fr

Abstract. In this paper, we are interested in the algorithmic complexity of computing (dis)similarity measures between two genomes when they contain duplicated genes. In that case, there are usually two main ways to compute a given (dis)similarity measure M between two genomes G_1 and G_2: the first model, that we will call the *matching model*, consists in computing a one-to-one correspondence between genes of G_1 and genes of G_2, in such a way that M is optimized in the resulting permutation. The second model, called the *exemplar model*, consists in keeping in G_1 (resp. G_2) exactly *one* copy of each gene, thus deleting all the other copies, in such a way that M is optimized in the resulting permutation. We present here different results concerning the algorithmic complexity of computing three different similarity measures (number of common intervals, MAD number and SAD number) in those two models, basically showing that the problem becomes **NP**-complete for each of them as soon as genomes contain duplicates. In the case of MAD and SAD, we actually prove that, under both models, both MAD and SAD problems are APX-hard.

1 Introduction

A classical method to compute phylogenies based on gene orders in whole genomes is to first compute a pairwise distance matrix, obtained by comparing every pair of genomes in a considered dataset. Note that the pairwise distance between two gene orders G_1 and G_2 is often deduced from a similarity or dissimilarity measure, say M, between G_1 and G_2. Gene orders are usually represented by signed *sequences* on the alphabet of *gene families*, where every element in a

* Work partially supported by the French-Italian Galileo Project PAI 08484VH and the French-Québec 60th CPCFQ.

genome is a *gene* (each gene being represented either as a letter or an integer). Any gene belongs to a unique gene family, and two genes that have the same label, regardless of the sign, belong to the same gene family. A family f of genes is said to be *balanced* between two genomes if the number of occurrences of genes of f is the same in both genomes. Two genomes G_1 and G_2 are consequently said to be *balanced* if all families of genes in G_1 and G_2 are balanced. In this paper we consider only balanced pairs of genomes, for reasons that we describe later in this section. When there are no duplicates in a balanced pair of genomes, it can be represented by a signed permutation, and the computation of the measure M is polynomial-time solvable in several models that have been used in practice (e.g. number of breakpoints, reversal distance, number of conserved intervals, number of common intervals, MAD and SAD numbers among others). However, aside of a few exceptions, like small genomes of viruses and organelles, duplicated genes and other genomic markers are common in genomes, due to several evolutionary mechanisms, like gene duplication or whole genomes duplications [San01]. Hence, it is an important problem to be able to handle those duplicates when measuring the distance or (dis)similarity between pairs of genomes. The main approach is to transform the two signed sequences G_1 and G_2 in a signed permutation, that allows to use one of the models defined in this case. This has been done in two different ways, called the *matching* model and the *exemplar* model. In the *matching* model, we want to end up in a situation where the pair of genomes is balanced. For this, for each family f of genes, we establish a maximal one-to-one correspondence between the genes of f in G_1 and the genes of f in G_2, before deleting the genes of f which have not been paired. The result is a complete matching between G_1 and G_2 that induces a signed permutation. The goal is to compute the matching that induces a signed permutation that is optimal, among all signed permutation given by all possible matchings, for the chosen measure M. In the *exemplar* model, introduced by Sankoff [San99], the idea is to keep exactly *one copy* of each family in each genome, that can be seen as the ancestral copy of all occurrences of the corresponding gene family. Consequently, we end up here again with a signed permutation and the goal is to compute the exemplar copy of each gene family that induces a signed permutation that is optimal for M. Let $occ(G, g)$ denote the maximum number of occurrences of a gene g in genome G (regardless of the signs), and let $occ(G)$ be the maximum of $occ(G, g)$ over all genes g in G. Note that if $occ(G_1) = 1$ (that is, if G_1 contains no duplicates), then for any genome G_2, both the matching and the exemplar models coincide.

In this paper, we present new results on the algorithmic complexity of computing different (dis)similarity measures, for balanced[1] pairs of genomes that

[1] Note that comparing real genomes often lead to unbalanced pairs of signed sequences and requires either models of measure that accounts for the insertion and deletion of gene and gene segments [BCF04] or methods to balance a pair of genomes [CZF$^+$05]. However, for all models we are aware of, insertion/deletions are not necessary for a balanced pair of genomes, and then the complexity of the balanced problem is sufficient to give the complexity of the more general problem.

contain duplicates. Several measures have already been studied like the number of breakpoints [Bry00,BCF04], the number of reversals [Bry00,CZF+05] and the number of conserved intervals [BR05]. Basically, it has been shown that, for each of the above mentioned measures, the problem is **NP**-complete as soon as duplicates are present in genomes, in both the matching and exemplar approaches ; some inapproximability results are also given in some cases [Tha05]. Here, we follow the same line and we describe results for three other similarity measures, namely number of common intervals, Maximum Adjacency Disruption number (MAD) and Summed Adjacency Disruption number (SAD), which will be defined in Section 2. In order to simplify notations, and since none of those three measures depends on the signs given to the genes (unlike measures like breakpoints, conserved intervals and reversals), we consider only *unsigned genomes*. We focus in Section 3 on the problem of computing the number of common intervals in genomes containing duplicates, and show that the problem is **NP**-complete in both the matching and exemplar models. In Sections 4 and 5, we prove that, under both models, both the MAD and SAD problems are APX-hard when genomes contain duplicates. Due to space constraints, no formal proof will be given in this extended abstract ; they will appear in the journal version of the paper.

2 Preliminaries

In this section, we define the three similarity measures we are interested in. As mentioned before, each of those measures asks that the genomes are balanced, and that a one-to-one correspondence exists between any gene of G_1 and a gene of G_2. Hence we will often conveniently rename genome G_1 into the identity permutation on n genes, Id_n (that is, $1\ 2\ 3 \ldots n$) and G_2 can be recomputed accordingly into a new permutation. We now define those three measures.

Number of common intervals: a *common* interval between G_1 and G_2 is a substring of G_1 for which the exact same content can be found in a substring of G_2. For example, let $G_1 = Id_5$ and $G_2 = 1\ 5\ 3\ 4\ 2$, then the interval $[3, 5]$ of G_1 is a common interval.

Maximum Adjacency Disruption Number (MAD): this notion has been recently introduced by Sankoff and Haque [SH05], where a genome is represented by a string of integers. This number, say $\mathcal{M}$, is defined as the maximum between two values $M_{1,2}$ and $M_{2,1}$, where $M_{1,2}$ (resp. $M_{2,1}$) is the maximum difference between two consecutive genes (i.e., integers) in G_2 (resp. G_1) , supposing that $G_1 = Id_n$ (resp. that $G_2 = Id_n$) and that G_2 (resp. G_1) has been renamed accordingly. We need to compute both $M_{1,2}$ and $M_{2,1}$ in order to restore symmetry, since those two measures might differ.

Summed Adjacency Disruption Number (SAD): this notion has also been introduced by Sankoff and Haque [SH05], and can be seen as a global variant of

the MAD number. Similarly to the previous case, suppose $G_i = Id_n$ and $G_j = g_1^j\ g_2^j \dots g_n^j$ has been renamed accordingly ($i \neq j \in \{1, 2\}$). The Summed Adjacency Disruption number is then defined as $\mathcal{S} = \sum_{k=1}^{n-1} |g_k^1 - g_{k+1}^1| + \sum_{k=1}^{n-1} |g_k^2 - g_{k+1}^2|$. In other words, we sum the differences between consecutive genes, and we do that in both "directions" to avoid asymmetry.
Note that the two last measures are actually *dissimilarity* measures, which means that the goal is to minimize them, while the first is a similarity measure that we wish to maximize.

3 Number of Common Intervals

Let EComI (resp. MComI) denote the problem of computing the maximum number of common intervals in the exemplar (resp. matching) model. We show that both EComI and MComI are **NP**-complete, even for restricted instances. The reduction we give below is valid for both models, since it shows NP-completeness in the case $occ(G_1) = 1$. However, in order to simplify notations, we will mention here only the exemplar model (i.e., the EComI problem). The reduction is from VertexCover. Starting from any instance of VertexCover (that is, a graph $G = (V, E)$ with $V = \{v_1, v_2 \dots v_n\}$ and $E = \{e_1, e_2 \dots e_m\}$), we will first describe a polynomial-time construction of two genomes G_1 and G_2 such that $occ(G_1) = 1$ and $occ(G_2) = 2$. We first describe G_1: $G_1 = b_1,\ b_2 \dots b_m,\ x,\ a_1,\ C_1,\ a_2,\ C_2 \dots a_n,\ C_n,\ y,\ b_{m+n},\ b_{m+n-1} \dots b_{m+1}$

The a_is, the b_is, x and y are genes, while C_is are sequences of genes. They are defined as follows:

- for any $1 \leq i \leq n$, $a_i = 2(i-1)m + i$;
- for any $1 \leq i \leq n$, $C_i = (a_i + 1),\ (a_i + 2) \dots (a_i + 2m)$;
- for any $1 \leq i \leq n + m$, $b_i = a_n + 2m + i$;
- $x = b_{n+m} + 1$;
- $y = b_{n+m} + 2$.

It can be seen that no gene appears more than once in G_1, thus $occ(G_1) = 1$. Now we describe the construction of G_2:
$G_2 = y,\ a_1,\ D'_1,\ b_{m+1},\ a_2,\ D'_2,\ b_{m+2} \dots b_{m+n-1},\ a_n,\ D'_n,\ b_{m+n},\ x$

The duplicated genes in G_2 are $b_1, b_2 \dots b_n$, and are spread within the D'_is. Moreover, each b_i, $1 \leq i \leq n$ will appear only twice in G_2. We now describe the contents of D'_i, $1 \leq i \leq n$. Basically, D'_i is constructed in two steps: (1) we first construct, for each i, a sequence of genes D_i, which is a specific shuffle of the contents of $C_i = (a_i + 1),\ (a_i + 2) \dots (a_i + 2m)$. More precisely, let $\min = a_i + 1$ and $\max = a_i + 2m$; then $D_i = (a_i + 3),\ (a_i + 5) \dots (a_i + 2m - 3),\ (a_i + 2m - 1),\ \min,\ \max,\ (a_i + 2),\ (a_i + 4) \dots (a_i + 2m - 4),\ (a_i + 2m - 2)$; (2) for any $1 \leq i \leq n$, we obtain D'_i by adding some b_js ($1 \leq j \leq m$) into D_i, accordingly to the initial graph G we are given. More precisely, for any edge e_j that is incident to a vertex v_i in G, we add the gene b_j between the j-th and the $(j+1)$-th gene of D_i. This process gives us the D'_is. Note that no two b_js ($1 \leq j \leq m$) can appear contiguously in a D'_i, and that no D'_i starts or ends with a b_j (all D'_is start and end with a gene that only appears in C_i in G_1).

Lemma 1. *Let G be a graph and G_1 and G_2 be the two genomes obtained by the construction described above. G admits a Vertex Cover VC such that $|VC| \leq k$ iff there exists an exemplar genome G_2^E obtained from G_2 having at least $\mathcal{I} = 2nm + 4n + m + 3 - 2k$ common intervals.*

As a direct consequence of Lemma 1, we conclude that the EComI problem is **NP**-complete. Moreover, as mentioned before, the proof and the result are also valid for the MComI problem, since our construction implies $occ(G_1) = 1$. We thus have the following theorem.

Theorem 1. *The* EComI *and* MComI *problems are both* **NP**-*complete, even when $occ(G_1) = 1$ and $occ(G_2) = 2$.*

We also consider, for the matching model, instances for which the constraints do not rely on the maximum number of duplicates per family, but on the number of families that contain duplicates. We have the following result.

Theorem 2. *The* MComI *problem is* **NP**-*complete, even when $f(G_1) = f(G_2) = 1$, where $f(G)$ denotes the number of different families of genes that contain duplicates in G.*

4 Maximum Adjacency Disruption (MAD)

Let EMAD (resp. MMAD) denote the problem of computing the minimum MAD number of in the exemplar (resp. matching) model. In this section, we prove inapproximability results for both the EMAD and MMAD problems. More precisely, we show that for no $\varepsilon > 0$, EMAD (resp. MMAD) admits a $(2 - \varepsilon)$-approximation algorithm, unless P=NP. This inapproximability result does not rely on the PCP theorem. We will also remark however, how, reconsidering the reduction proposed in view of APX-hardness results based on the PCP theorem, one can replace the constant 2 above with a strictly bigger constant. The proof is split into two: we first study the complexity of a restricted form of SAT, which we call Uniform-SAT, and in particular we show that it is **NP**-complete. Next, we show that a $(2-\varepsilon)$-approximation algorithm for EMAD (resp. MMAD), for some $\varepsilon > 0$, would imply the existence of a polynomial-time algorithm for Uniform-SAT. Finally, we obtain the inapproximability result for EMAD (resp. MMAD).

In the following, 3SAT will denote the restriction of SAT for which each clause contains at most 3 literals. We introduce a restricted form of 3SAT called Uniform-SAT, as follows: an instance $\langle X, \mathcal{C} \rangle$ of 3SAT is an instance of Uniform-SAT when the following two conditions are met: (i) for each clause $C \in \mathcal{C}$, either all literals occurring in C are positive occurrences of variables from X or all literals occurring in C are negated occurrences of variables from X and (ii) for each variable $x \in X$, x has at most 3 positive and at most 2 negated occurrences within $\mathcal{C}$. A 3SAT formula $F = \bigwedge_{C \in \mathcal{C}} C$ is called 3-*bounded* if no variable has more than 3 occurrences within C and is called $(2,2)$-*bounded* if no variable has more than 2 positive occurrences and no more than 2 negated occurrences within C. The following two facts are known: (1) the decision problem 3SAT is **NP**-complete even when restricted to 3-bounded formulas and (2) the optimization problem MAX-3SAT is APX-hard even when restricted to 3-bounded formulas [GJ79]. Since both problems admit a trivial self-reduction in case a variable has only positive (or only negated) occurrences, then the following two facts also hold: (1) 3SAT is **NP**-complete even when restricted to $(2,2)$-bounded formulas and (2) MAX-3SAT is APX-hard even when restricted to $(2,2)$-bounded formulas. Notice that, of the above two results, only the second is related to the PCP-theorem.

Theorem 3. *Deciding whether a given* UNIFORM-SAT *formula is satisfiable is* **NP***-complete.*

Theorem 3 here above does not need the PCP theorem and is all what is required in the following for proving that, for no $\varepsilon > 0$, EMAD (resp. MMAD) admits a $(2-\varepsilon)$-approximation algorithm, unless P=NP. With dependence on PCP, we have the following result, which, besides being of independent interest, can be used to show that the right constant for the approximability of EMAD (resp. MMAD) is not 2.

Theorem 4. *Given a* UNIFORM-SAT *formula, the problem of finding a truth assignment maximizing the number of satisfied clauses is APX-hard.*

We now prove that both the EMAD and MMAD problems are APX-hard. The result holds for both problems, since we prove it in the case where $occ(G_1) = 1$, where they coincide. The result rests on a reduction form UNIFORM-SAT. Assume we are given an instance $\langle X, \mathcal{C} \rangle$ of UNIFORM-SAT, where $X = \{x_1, x_2, \ldots, x_n\}$. Here, $\mathcal{C}$ can be partitioned into the family $\mathcal{P} = \{P_1, P_2, \ldots, P_{m_p}\}$ of clauses comprising only positive literals and the family $\mathcal{N} = \{N_1, N_2, \ldots, N_{m_n}\}$ of clauses comprising only negated literals. Let M_ε be a sufficiently big positive integer that we will fix later in order to force our conclusions. We propose to compare two genomes G_1 and G_2. Here, G_1 is the simple (that is, without repetitions) genome G_1 of length $L_1 = 2\,M_\varepsilon + m_p + m_n + n - 1$ defined as follows: $G_1 = 1\ 2\ 3 \ldots L_1$. A gene at position i in G_1 with $i \leq m_p$ or $i \geq L_1 - m_n + 1$ is called a $*$-*gene*. Genome G_2 has length $L_2 = 2\,M_\varepsilon + 6\,n - 1$, and conforms to the following pattern, where we have found it convenient and pertinent to spot out the displacement of the $*$-genes within genome G_2.

$$G_2 = m_p + 1, \ldots, m_p + M_\varepsilon, *, *, *, *, *, m_p + M_\varepsilon + 1, *, *, *, *, *, m_p + M_\varepsilon + 2, \ldots$$
$$\ldots, *, *, *, *, *, m_p + M_\varepsilon + n, m_p + M_\varepsilon + n + 1, m_p + M_\varepsilon + n + 2, \ldots, m_p + 2\,M_\varepsilon + n - 1$$

We will specify later the precise identity of the $*$-genes within genome G_2. For now, notice that in G_2 we have precisely n runs of 5 consecutive $*$-genes. We put these runs into $(1,1)$-correspondence with the n variables in X, so that the i-th run corresponds to variable x_i, for $i = 1, 2, \ldots, n$. For each $i = 1, 2, \ldots, n$, let

$\mathcal{P}_i$ and $\mathcal{N}_i$ be the lists of index sets of the clauses from $\mathcal{P}$ and $\mathcal{N}$ which contain variable x_i. E.g., if x_i appears in P_3, in P_7, and in N_2, then $\mathcal{P}_i = (3, 7)$, whereas $\mathcal{N}_i = (2)$. Notice that the lengths of the lists $\mathcal{P}_i$ and $\mathcal{N}_i$ are at most 3, and 2, respectively. ¿From the list $\mathcal{P}_i$ we obtain a list $\mathcal{P}'_i$ of length precisely 3 by possibly iterating the last element in $\mathcal{P}_i$ the required number of times (that is, $3 - |\mathcal{P}_i|$ times). A list $\mathcal{N}'_i$ of length precisely 2 is similarly obtained from list $\mathcal{N}_i$. Now, for each $i = 1, 2, \ldots, n$, the i-th run of 5 consecutive $*$-genes consists in the following 5 characters: $(*, *, *, *, *) \to (\mathcal{P}'_i[1], \mathcal{P}'_i[2], \mathcal{P}'_i[3], L_1 - m_n + \mathcal{N}'_i[1], L_1 - m_n + \mathcal{N}'_i[2])$

The above reduction leads us to the following result.

Theorem 5. *For no $\varepsilon > 0$,* EMAD *(resp.* MMAD*) admits a $(2 - \varepsilon)$-approximation algorithm, unless P=NP.*

Remark 1. There actually exists a constant $c > 2$ such that EMAD (resp. MMAD) admits no c-approximation algorithm unless P=NP. We can get to this stronger conclusion if in the proof of Theorem 5 here above we apply Theorem 4 instead of Theorem 3.

5 Summed Adjacency Disruption (SAD)

Let ESAD (resp. MSAD) denote the problem of computing the minimum SAD number of in the exemplar (resp. matching) model. In this section, we prove that both problems ESAD and MSAD, expressed on two genomes G_1 and G_2 such that $|G_1| \leq |G_2|$ cannot be better than $\log(|G_1|)$ approximated. This result holds for both the exemplar and the matching models, since we prove it in the case where $occ(G_1) = 1$, for which the two problems coincide. The inapproximability of ESAD (resp. MSAD) is obtained starting from the inapproximability of SetCover. This result will hence depend on the PCP theorem, but will deliver stronger SetCover-like inapproximability thresholds than for the EMAD ans MMAD problems discussed in the previous section.

Let $\langle V, \mathcal{S} \rangle$ be an instance of SetCover, where $V = \{1, 2, \ldots, n\}$, and $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ is a family of subsets of V. We can assume n is even, say $n = 2k$, and each set S_i contains precisely $k = \frac{n}{2}$ elements, say $s^i_1, s^i_2, \ldots, s^i_k$. The well-known inapproximability results for SetCover hold also under these assumptions, since we can think of enlarging a ground set V, originally on k elements, by adding a set V' of k new elements, adding V' to $\mathcal{S}$, and enlarging the other sets in $\mathcal{S}$ with elements from V' until their size rises up to k. Let $M = m^3 n^3$ play the role of a sufficiently big positive integer. We propose to compare two genomes G_1 and G_2. Here, G_1 is the simple genome G_1 of length $L_1 = M + n + m$ defined as follows: $G_1 = 1\ 2\ 3 \ldots L_1$. Genome G_2 has length $L_2 = M + m(k + 1)$, and is constructed as follows:

$$G_2 = n{+}1, n{+}2, \ldots, n{+}M, s^1_1, s^1_2, \ldots, s^1_k, n{+}M{+}1, s^2_1, s^2_2, \ldots, s^2_k, n{+}M{+}2, \ldots$$
$$\ldots, s^{m-1}_1, s^{m-1}_2, \ldots, s^{m-1}_k, n + M + m - 1, s^m_1, s^m_2, \ldots, s^m_k, n + M + m$$

The above reduction leads us to the following result.

Theorem 6. *There exists a constant $c > 0$ such that* ESAD *(resp.* MSAD*) admits no ($c \log |G_1|$)-approximation algorithm, unless P=NP.*

6 Conclusion

In this paper, we have investigated the algorithmic complexity of the problem of computing similarity measures between genomes, in the case where genomes contain duplicates. This has been done for three measures: common intervals, MAD and SAD. We have shown that the three problems are **NP**-complete, for both the exemplar and matching variants. Moreover, we have provided APX-hardness results concerning MAD and SAD. Our results basically show that as soon as duplicates are present, the problems become hard, even in very restricted instances. Moreover, as can be seen, no APX-hardness result is known concerning common intervals ; we are currently investigating those questions.

References

[BCF04] G. Blin, C. Chauve, and G. Fertin, *The breakpoint distance for signed sequences*, 1st Int. Conference on Algorithms and Computational Methods for Biochemical and Evolutionary Networks, CompBioNets 2004, Texts in Algorithms, vol. 3, KCL Publications, 2004, pp. 3–16.

[BR05] G. Blin and R. Rizzi, *Conserved interval distance computation between non-trivial genomes*, 11th Int. Comp. and Combinatorics Conference (COCOON'05), LNCS, vol. 3595, 2005, pp. 22–31.

[Bry00] D. Bryant, *The complexity of calculating exemplar distances*, Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families (D. Sankoff and J. Nadeau, eds.), Kluwer Acad. Pub., 2000, pp. 207–212.

[CZF+05] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang, *Assignment of orthologous genes via genome rearrangement*, IEEE/ACM Trans. on Comp. Biology and Bioinformatics **2** (2005), no. 4, 302–315.

[GJ79] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman, San Franciso, 1979.

[San99] D. Sankoff, *Genome rearrangement with gene families*, Bioinformatics **15** (1999), no. 11, 909–917.

[San01] ———, *Gene and genome duplication*, Curr. Opin. Genet. Dev. **11** (2001), no. 6, 681–684.

[SH05] D. Sankoff and L. Haque, *Power boosts for cluster tests*, Comparative Genomics, RECOMB 2005 International Workshop, RCG 2005, LNBI, vol. 3678, Springer, 2005, pp. 121–130.

[Tha05] N. Cam Thach, *Algorithms for calculating exemplar distances*, Honours Year Project Report, National University of Singapore, 2005.