Antrag an die Deutsche Forschungsgemeinschaft
Kennedyallee 40, 53175 Bonn

auf Gewährung einer Sachbeihilfe

Dual Sparse De-Bruijn Subgraphs (DSDS) for High-throughput Sequencing

Prof. Dr. Jens Stoye
Arbeitsgruppe Genominformatik

Technische Fakultät der Universität Bielefeld

und

Institut für Bioinformatik
Centrum für Biotechnologie der Universität Bielefeld

1 General information (Allgemeine Angaben)

Proposal for a research grant (new application).

1.1 Applicant (Antragsteller)

Name:	Prof. Dr. Jens Stoye
Employment status:	Universitätsprofessor (C4)
Date of birth:	17. 03. 1970
Nationality:	german
Institution:	Technische Fakultät und Institut für Bioinformatik der Universität Bielefeld
Office address:	Universität Bielefeld Technische Fakultät AG Genominformatik 33594 Bielefeld
Telephon:	0521 - 106 - 3852
Telefax:	0521 - 106 - 6495
E-Mail:	stoye@techfak.uni-bielefeld.de
Private address:	Kesselstraße 3 33602 Bielefeld 0521 – 9675479

See attached documents for a tabular CV.

1.2 Topic (Thema)

Watson-Crick complement invariant version of sparse De-Bruijn subgraphs for analysis of high-throughput sequencing data.

Watson-Crick-Komplement-unabhängige Version von sparse De-Bruijn-Teilgraphen für die Analyse von Hochdurchsatz-Sequenzierdaten.

1.3 Scientific discipline and field of work (Fachgebiet und Arbeitsrichtung)

Scientific discipline: Computer Science, algorithm design and engineering
Field of work: Sequence analysis

1.4 Scheduled total duration (Voraussichtliche Gesamtdauer)

Duration: 30 months

1.5 Application period (Antragszeitraum)

First proposal: Yes
Time period: 1st January 2008 – 30th June 2010
Funding commence: 1st January 2008

1.6 Summary (Zusammenfassung)

High-throughput DNA sequencing methods in use today are able to produce staggering amounts of data on a daily basis that demand extensive computing resources to assemble, finish and process genomic information. The Center for Biotechnology (CeBiTec) at Bielefeld University will soon have its own high-throughput sequencing machine, a patented technology developed by 454 Life Sciences, which is capable of producing millions of raw DNA bases per hour. However, what makes sequencing a challenge today is the subsequent computational problem of correctly assembling the reads into the original sequence.

This project aims to develop a novel approach to genome assembly based on Dual Sparse De-Bruijn Subgraphs (DSDS). De-Bruijn subgraphs are already being studied by members of our research group. The new approach consists of a version of our current model that exploits the Watson-Crick complementarity nature of the DNA. The project consists of two parts. First, we will study and develop the new graph, and design the necessary data structures for efficient implementation. In the second part of the project, we will apply our new approach in practice to genomic assembly of DNA data sequenced at the CeBiTec and other institutions and, extending ongoing research, to detect repeats in these data.

Zusammenfassung

Hochdurchsatztechniken, wie sie mittlerweise immer mehr Verbreitung in der DNA-Sequenzierung finden, erlauben die Produktion atemberaubend großer genomischer Sequenzdatenmengen in kurzer Zeit, was hohe Anforderungen auch für die bioinformatische Auswertung mit sich bringt. Das Zentrum für Biotechnologie (CeBiTec) der Universität Bielefeld wird in Kürze eine eigene Hochdurchsatz-Sequenziermaschine der Firma Roche/454 in Betrieb nehmen. Die große Herausforderung beim Einsatz dieser Technologie ist allerdings die nachfolgende bioinformatische Analyse, insbesondere die Assemblierung der gelesenen Teilsequenzen in das ursprüngliche Genom.

Ziel des hier beantragten Projekts ist die Entwicklung einer neuen Methode zur Genomassemblierung auf der Basis von *Dual Sparse De-Bruijn Subgraphs* (DSDS). De-Bruijn-Teilgraphen werden bereits in unserer Arbeitsgruppe untersucht. Neuartig an dem hier vorgeschlagenen Ansatz ist eine Erweiterung des Modells, das die Watson-Crick-Komplementarität von DNA-Sequenzen berücksichtigt. Das Projekt besteht aus zwei Teilen. Im ersten Teil soll die neue Graph-Datenstruktur definiert und theoretisch untersucht sowie effizient implementiert werden. Im zweiten Teil soll der neue Ansatz in der Praxis erprobt werden, zum einen bei der Assemblierung genomischer DNA und zum anderen in Ergänzung bestehender Forschungen bei der *Repeat*-Analyse dieser Daten.

2 State of the art, preliminary work (Stand der Forschung, eigene Vorarbeiten)

2.1 State of the art (Stand der Forschung)

De-Bruijn graphs were first studied in the end of the 19th century, although in an implicit form, and were formally defined in 1946 by N. G. de Bruijn [20, Chapter 3]. They are directed graphs with a simple and clear definition that are easy to build, and that have interesting properties: they are regular, have small diameter, and are both Hamiltonian and Eulerian. As a result, De-Bruijn graphs have been used in diverse applications such as network models, pseudo-random number generation, and DNA analysis [20]. Similar graphs have also been used for the identification of repeat families [27], but the first use in Bioinformatics was probably the Eulerian path approach to sequence assembly proposed by Idury and Waterman [17] and extended by Pevzner, Tang and Waterman [23].

Despite the success achieved by the sequence assembler developed by Pevzner and co-workers for bacterial genomes, De-Bruijn graphs have not been further explored in computational biology. In some works they appear slightly modified, and are more often used as a multiple alignment displayer than a basic tool [22, 27]. As Myers [21] points out, the main problem with De-Bruijn graphs is that, in the original definition, they are space inefficient.

A d-dimensional De-Bruijn Graph $G = (V, E)$ on an alphabet Σ is a directed graph defined as follows:

$$\begin{aligned} V &= \Sigma^d \\ E &= \{(u, v) \mid u, v \in V \text{ and } u_{i+1} = v_i, \text{ for all } 1 \leq i < d\}, \end{aligned}$$

where u_i is the i-th character of the string represented by u. Strings of length at least d over the same alphabet describe walks on the d-dimensional De-Bruijn graph. Given a set of strings, we define the associated d-dimensional *De-Bruijn subgraph* as the subgraph of the d-dimensional De-Bruijn graph that contains all the walks described by these strings but no extra vertex or arc.

Sequence-associated De-Bruijn subgraphs have a nice asymptotic behavior: The maximum number of nodes increases linearly with the size of the input and decreases with the dimension d of the graph. The main problem is that, although these graphs scale well with the string set size, graphs corresponding to even relatively small genomes (such as of bacteria) are already prohibitively large for an average computer. This could probably explain why De-Bruijn graphs have not been further used in Bioinformatics.

Another disadvantage of simple De-Bruijn graphs for genome assembly is that they require cutting the sequence reads into small pieces to finally build the graph. Myers [21] suggested that this step might not be necessary. In fact, we have successfully constructed a hybrid structure based on De-Bruijn subgraphs without explicitly using sequence comparisons, with the added advantage of allowing the representation of a whole genome of a higher-order organism in the memory of a typical personal computer.

In a typical sequence-associated De-Bruijn subgraph, the graph branches are more important than the nodes in non-branching paths, and most of the nodes in such a graph are exactly in non-branching paths. Hence, the first step to reduce its size is to contract non-branching paths into single nodes containing sequences of unlimited length, similarly to what Idury and Waterman [17] have done in their assembly algorithm.

In order to avoid constructing the large non-contracted graph, we define an indexed structure, called *sequence graph*, formed by a relaxed definition of a De-Bruijn graph and a mapping from d-tuples to the nodes that represent them. In the relaxed De-Bruijn graph, the length of a sequence in a node is unbounded. This allows us to create a compact representation of sparse De-Bruijn subgraphs without non-branching paths, while the access to a particular d-tuple is still possible via the index. Our initial experiments show that the sequence graph is able to store the same information as a De-Bruijn subgraph using roughly 25% of the memory. We also notice a proportional reduction of the time spent in constructing the graph.

We believe that the sequence graph can be further improved by exploiting a peculiarity of genomic data, namely, the reverse-complementarity nature of DNA. A DNA molecule is formed by two complementary strands that bind according to pairing rules in the double-helix structure discovered by Watson and Crick, namely guanine (G) pairs with cytosine (C), and adenine (A) pairs with thymine (T). Current sequencing methods are not able to distinguish from which strand a given sequence comes and, as a result, each sequence read is represented twice in the sequence graph. Our aim is to avoid this duplication by adding the concept of complementarity both in the index and in the relaxed De-Bruijn graph. The structure we propose is called Dual Sparse De-Bruijn Subgraph. In case such a structure can be efficiently built and updated, we will be able to double the efficiency of the structure we already have both in terms of memory requirements and construction time.

2.2 Preliminary work, progress report (Eigene Vorarbeiten, Arbeitsbericht)

The proposed data structure has several applications in genome analysis, which is the main subject of most of the activities in our Genome Informatics research group, as can be seen below.

Sequence Searching. The group worked in close collaboration with the research group for Practical Computer Science in Bielefeld, headed by Prof. Robert Giegerich, on indexing methods for large amounts of genomic data. Among them are the *suffix trees* [9], which are at the moment very popular in the field of Bioinformatics. Suffix trees can be used in many different applications [10]. The identification of repeated sequences [11, 18, 32] is one of them.

Affix trees [31] are a generalization of suffix trees for applications related to the search for a given pattern in a text. They represent a connection between suffix trees and their corresponding complementary reverse *prefix trees*. Each substring of a text is therefore represented twice (as if we read it from both left to right and right to left). This gives support to flexible search strategies. The linear memory requirement of affix trees is known since its development in Bielefeld in 1995. An efficient algorithm for constructing them was presented later by Maaß [19].

While suffix trees, and specially affix trees, theoretically allow the implementation of both time and space efficient algorithms, the amount of memory required by such structures in practice are not negligible. An efficient alternative is the use of *suffix arrays*. They can be used in almost all scenarios where suffix trees can be applied, and their representation requires much less memory in practice [1]. An algorithm for the construction of suffix arrays from genomic data was developed in the Genome Informatics research group [30]. The algorithm may be used in many different applications, and has a running-time comparable to the best existing algorithms, being even faster in many cases.

An even less memory consuming index for database searches is the so called *q-gram hash*, where the positions of all q-grams (substrings of length q) are stored in a table. Based on this index, we were able to design a filter algorithm in collaboration with Gene Myers (Janelia Farm Research Campus of the Howard Hughes Medical Institute). Besides the filter algorithm, the database search software tool SWIFT [28] was also a result of this collaboration. SWIFT not only requires much less memory than similar software tools, but is also 25 times faster than the most popular of them, BLAST, without any loss of sensitivity.

Genome evolution modeling. Information about the evolution of genomes may be obtained from the global structure of the gene arrangements in them. In this case, the corresponding maximum parsimony assumption, namely the minimum number of rearrangements (inversions, translocations, transpositions, fusions, and fissions of genome pieces), is used to transform a genome into another one. The related mathematical theory was studied by Hannenhalli and Pevzner [13, 14, 12] during the 1990's. The highly technical content of their work was at that time a big barrier to the area. Together with Anne Bergeron (Montreal), we could make important steps to the simplification of this theory [2, 3, 4, 6]. Errors in the original works could also be identified and corrected during the simplification process [5]. Another object studied by our group are gene clusters, which are sets of genes that appear close to each other in different, not so closely related genomes. In [15, 16, 29] we developed many different algorithms able to efficiently identify gene clusters in several genomes.

Systems biology and microarray design. Systems biology explores biological systems and their interactions both mathematically and algorithmically. One of our aims is the development of methods for the improvement of data quality, which ideally could be applied just after or even during the data collection, as well as methods for the integration of data from different experiments. In collaboration with the chair of Genetics in Bielefeld, members of our group analyzed the regulatory network of *Corynebacterium glutamicum*. Another topic of our research is the design of microarrays, which consists of optimizing the selection of probe sequences [24, 26], reducing the length of the synthesis schedule [25], and designing the layout of the array (the arrangement of the probes on the chip) in order to improve the quality of the manufacturing process [7, 8].

Repeat analysis. Identifying repetitive elements in the genome of eukaryotes is an important task both if one is interested in studying them and if one needs to avoid them. Although the sequence of some repetitive elements may be characterized, which allows repeat identification by traditional sequence comparison methods, the experience shows that most of the eukaryotes have very specific families of repetitive elements. For completely sequenced genomes, the *de novo* identification of such sequences may be done using suffix trees. A software tool called REPuter [18] was developed based on this approach. In a collaboration with Prof. Bernd Weisshaar (chair of Genome Research, Bielefeld University), we are developing a De-Bruijn subgraph based method for doing the same with collections of reads of an incompletely sequenced genome. Also here the size of De-Bruijn subgraphs is a considerable barrier to the development of software tools for personal computers.

3 Goals and work schedule (Ziele und Arbeitsprogramm)

3.1 Goals (Ziele)

While working with a De-Bruijn based approach to repeat finding, we were able to develop an algorithm for building the sequence graph, the compact version of a De-Bruijn subgraph, directly

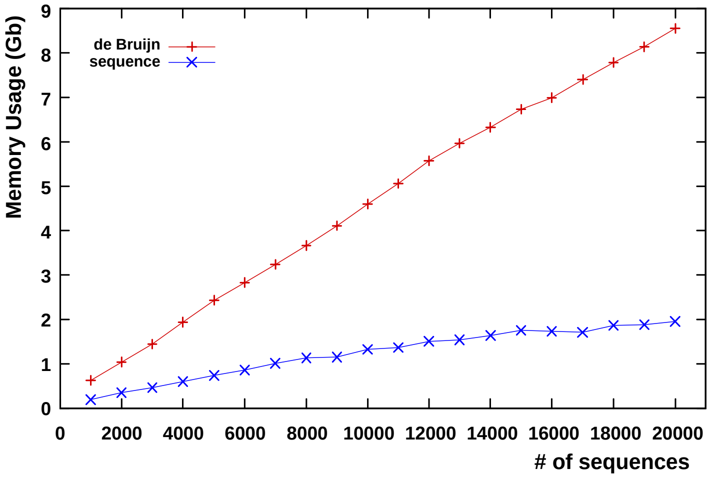


Figure 1: Memory usage (in gigabytes) for De-Bruijn subgraphs and their corresponding sequence graphs. The number of sequences corresponds to the number of 700 bases length randomly taken subsequences of an *Arabidopsis thaliana* chromosome.

from the given set of sequences. The graph in Figure 1 presents the amount of memory used by De-Bruijn graphs and by the corresponding sequence graphs. Figure 2 exemplifies how memory can be saved: The De-Bruijn subgraph containing both the sequences `AAACCC` and `TTGTGGG`, with the respective complements (Fig. 2.2), contains non-branching paths. Since the information we are interested in are exactly in the graph branches, the same information represented by the 16 nodes, 48 characters, and 14 edges (Fig. 2.2) can be represented by 2 nodes, 20 characters and no edge at all (Fig. 2.3).

Most bioinformatics applications need to deal with both the input set and its reverse complement. Without considering the reverse complement, important information can be lost, as Figure 2.1 shows. In this case, because only the input set is considered, one may conclude that the given sequences belong to independent DNA molecules, whereas Figure 2.2 shows that both sequences are indeed part of different strands of the same DNA molecule. The direct consequence of this is that in a De-Bruijn graph for the DNA alphabet, the information usually needs to be represented twice.

The goal of this project is to take advantage of the DNA sequence complementarity to reduce even more the graph size. In the example shown in Figure 2.4, the reduction achieved is of 50%. Of course this is a very special case. In the real life, we will be facing many problems not shown in the example, like self-complementary sequences.

More specifically, the goals of the project are:

(A) Investigation of the proposed Dual Sparse De-Bruijn Subgraph (DSDS) data structure.

(B) Implementation of a genome assembler, so that we can compare the performance of an application using the new graph to the performance of similar applications.

We choose the genome assembly problem for mainly three reasons:

- The De-Bruijn graph approach to genome assembly is well studied and documented. Not only the assembly, but also a sequencing error correction method are described in

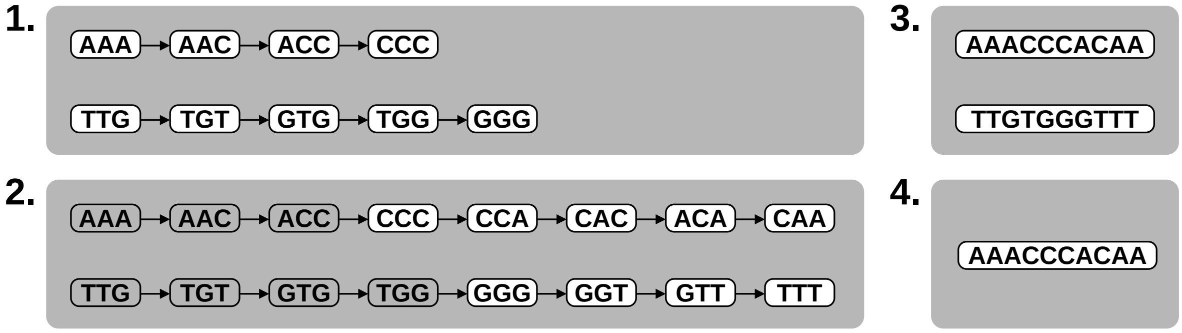


Figure 2: Four graphs for assembling the sequences AAACCC and TTGTGGG.

the literature.

- From the known applications for De-Bruijn graphs, the genome assembly is the only one with which the group has no previous experience. Since the application is well documented, it is a good chance to expand our group's knowledge without risking delaying the project.
- Although genome assembling may look like an old fashion subject, the appearance of completely new sequencing methods in the last years suggests that, in the near future, sequencing projects may involve lots of different kinds of sequences, with the most variable lengths. In this scenario, the De-Bruijn graph approach flexibility in respect of the input sequence lengths will be an advantage.

(C) Adaptation of the repeat detection method being developed in our group to use DSDS.

(D) Validation of the applications developed. For attaining this natural final goal, we count with collaborations both in Bielefeld and in Brazil. At the end, the project must produce, and publish, an implementation of DSDSs and two validated applications using them.

3.2 Work schedule (Arbeitsprogramm)

The project's work schedule consists of four phases, oriented towards the four goals described in the previous section. For a graphical overview, see the end of this section.

(A) Design

The first phase of the project consists of studying the properties of the Dual Sparse De-Bruijn Subgraph (DSDS) and designing the data structures on which will be built the genome assembler (phase B) and the integration with our existing repeat detection functions (phase C). In this phase, the storage requirements for the data structure will be studied in detail, so that the memory requirements for the genome assembler and repeat detection functions can be predicted based on the size of the input (sequencing data).

Among the functions the DSDS data structure shall offer, we point out:

InsertSequence - Makes the necessary transformations in the graph, so that the walk corresponding to a given sequence may be found in it.

DELETESEQUENCE - Removes the walk corresponding to a given sequence from the graph.

TRACESEQUENCE - Returns the nodes corresponding to a sequence walk in the order they appear in the sequence.

GETNODE - Returns the node representing a given q-gram.

In this phase a postdoctoral researcher will lead the design of the necessary data structures, with a team of two students starting the work on the basic DSDS implementation three months after the beginning of the design phase. Both the design and the DSDS implementation phases will continue in parallel as the implementation is likely to require changes in the design of the data structure. The design of basic data structures is expected to reach a stable state by October 2008 and a final version of the DSDS should be ready by the end of the year, although the first quarter of the second year may be also used in case any serious design problems arise during the implementation phase.

(B) Implementation

This phase consists of the implementation of the DSDS data structure as described above, and the implementation of a fully-functional genome assembler as a stand-alone software package.

The DSDS implementation will be carried out by a team of two student programmers lead by a postdoctoral researcher. An initial working implementation providing the basic functions should be ready by mid-2008 so that the DSDS design can be exhaustively evaluated and validated until the end of the first year using real sequencing data. A final implementation of the DSDS is expected to be ready by the end of 2008, or by the end of the first quarter of 2009 at the latest.

The implementation of the genome assembler using the DSDS data structure starts in 2009 with a single programmer working under the supervision of both postdoctoral researchers (with the other programmer working independently on phase C). The final version of the genome assembler should be ready by the end of 2009.

(C) Integration of existing repeat detection algorithm with DSDS

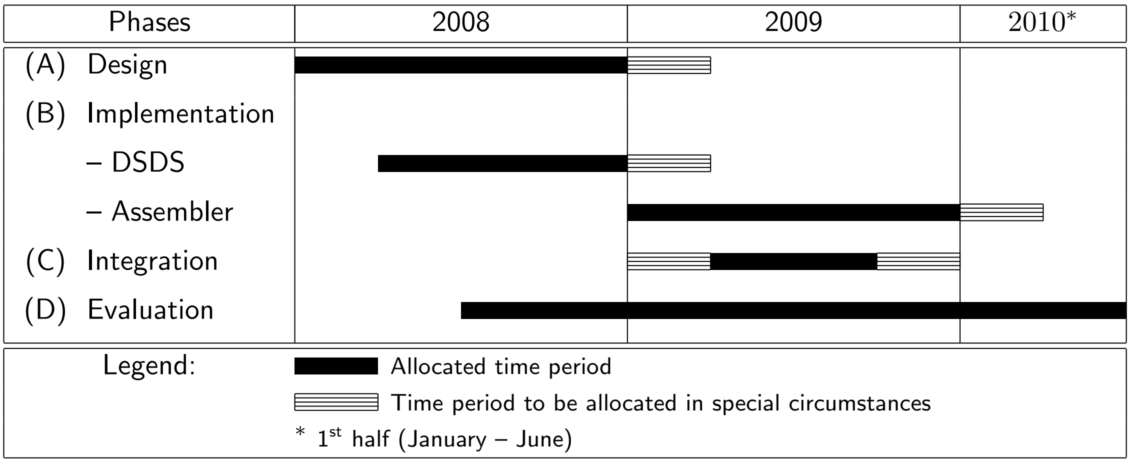
Once the implementation of the basic DSDS data structure becomes stable, it will be integrated with the existing repeat detection algorithms already developed at the Genome Informatics research group. This phase will be carried out by a single programmer under the supervision of both postdoctoral researchers, and it may be concluded in about six months, or at most nine months. This phase might start as early as January 2009. However, depending on the amount of work involved in the development of the genome assembler, it might be necessary to delay the integration phase by up to three months so that, in the beginning, both programmers can work on the genome assembler. As soon as the integration phase is concluded, both students will be free to concentrate on the implementation of the genome assembler, and also help with the evaluation (phase D).

(D) Evaluation

The DSDS implementation will be evaluated using the genome assembler built in phase B and the repeat detection algorithms integrated in phase C. This evaluation will use real sequencing data generated by the Center for Biotechnology (CeBiTec) in Bielefeld and our partners in Brazil. The validation of the results will be under the responsibility of both postdoctoral researchers, with the help of both students and biologists from the Genome Research group of Prof. Bernd Weisshaar and from the group of Dr. Felipe Rodrigues da Silva at Embrapa, in Brazil. Running times and memory requirements will also be measured in order to compare the results with those obtained with similar software packages.

The evaluation phase will start as soon as the initial implementation of the basic DSDS data structure is ready (mid-2008), and it will last until the end of the project, in parallel with the implementation and integration phases. The evaluation phase will also consist of publishing the results in journals and presenting them at international conferences.

Summary of work schedule



3.3 Experiments involving humans or humans materials (Untersuchungen am Menschen oder an vom Menschen entnommenem Material)

Not applicable.

3.4 Experiments with animals (Tierversuche)

Not applicable.

3.5 Experiments with recombinant DNA (Gentechnologische Experimente)

Not applicable.

4 Funds requested (Beantragte Mittel)

The following are the funds requested for this project for the period of 30 months.

4.1 Staff (Personalkosten)

The scientific staff requested for this project includes:

- 1 postdoctoral researcher (wiss. Mitarbeiter E14) for 18 months (Jan 2008 – June 2009)
- 1 postdoctoral researcher (wiss. Mitarbeiter E14) for 24 months (July 2008 – June 2010)
- 2 programmers (stud. Hilfskraft, 10 h/Woche) for 27 months each (Apr 2008 – June 2010)

The main research work of the project will be performed by two postdoctoral researchers, supported by two student programmers. The first postdoc will perform the initial design of the DSDS data structure (phase A), its implementation (phase B.1) and evaluation (phase D). The second postdoc will concentrate on the implementation of the assembler (phase B.2) and the integration of the repeat detection algorithm (phase C) as well as their evaluation (phase D). The student programmers will support the postdoctoral researchers, mainly in the implementation and evaluation phases.

A number of Ph.D. students will graduate in the next years from the International Graduate School in Bioinformatics and Genome Research and the GK Bioinformatik in Bielefeld, so we do not expect a shortage of highly qualified applicants for the postdoctoral positions. From the Bachelor and Master programs in *Naturwissenschaftliche Informatik* and *Bioinformatik und Genomforschung* at the Faculty of Technology of Bielefeld University there are more than enough well qualified candidates for the student programmer positions.

Summary:	1st Year (1. Jahr)	18 Months (Monate)	wiss. MA E14
(Zusammenfassung)		18 Months (Monate)	stud. HK, 10 h/Woche
	2nd Year (2. Jahr)	18 Months (Monate)	wiss. MA E14
		24 Months (Monate)	stud. HK, 10 h/Woche
	3rd Year (3. Jahr)	06 Months (Monate)	wiss. MA E14
		12 Months (Monate)	stud. HK, 10 h/Woche

4.2 Scientific instrumentation (Wissenschaftliche Geräte)

Using our current implementation of Sparse De-Bruijn Subgraphs, 96 GB of memory are sufficient for keeping any large genome entirely in memory. With the dual version, the memory requirements are expected to drop to about half of the current requirements and, hence, 48 GB of memory should suffice.

The Genome Informatics research group has acquired in early 2003 a compute server with 96 GB of memory, which is now part of the infrastructure provided by the Bioinformatics Resource Facility (BRF) of the CeBiTec. Although this machine has enough memory for running applications based on DSDS, it will be five years old by the time the project proposed here starts. Therefore, we request funds for acquiring a new compute server.

This Sun Fire X4600 M2 is equipped with eight 2.8 GHz AMD Opteron processors and has 64 GB of main memory. As argued above, this should suffice for the tasks planned in this project. The SUN architecture fits optimally in the existing infrastructure of the BRF, such that efficient set-up and maintenance are guaranteed.

Summary (Zusammenfassung):

1st Year (1. Jahr)	35.560,77 EUR
2nd Year (2. Jahr)	0,00 EUR
3rd Year (3. Jahr)	0,00 EUR
Sum (Summe) 4.2	**35.560,77 EUR**

4.3 Consumables (Verbrauchsmaterial)

No funds for consumables are requested. The materials needed in this project, including office material, will be provided from the budget of the Genome Informatics research group.

Summary (Zusammenfassung):

1st Year (1. Jahr)	0,00 EUR
2nd Year (2. Jahr)	0,00 EUR
3rd Year (3. Jahr)	0,00 EUR
Sum (Summe) 4.3	**0,00 EUR**

4.4 Travel (Reisen)

Although co-operation meetings with Dr. Felipe da Silva and his team are likely to be necessary during the course of the project, especially during the evaluation phase (beginning of 2nd year), no funds for these travels are requested, as these will be funded from other sources. Beyond that, one or two intercontinental or national/European conference attendances of the scientific researches are planned per year, as detailed in the following table:

wiss. MA E14	1st Year	1 Intercontinental conference	2.000,00 EUR
(Jan 2008 – June 2009)		1 European conference	1.500,00 EUR
	2nd Year	1 Intercontinental conference	2.000,00 EUR
	3rd Year		0,00 EUR
wiss. MA E14	1st Year		0,00 EUR
(Oct 2008 – June 2010)	2nd Year	1 Intercontinental conference	2.000,00 EUR
		1 European conference	1.500,00 EUR
	3rd Year	1 Intercontinental conference	2.000,00 EUR

Example of intercontinental conferences:
ISMB: *Conference on Intelligent Systems for Molecular Biology* (2008 in Toronto, Canada)
RECOMB: *Conference on Research in Computational Molecular Biology* (2008 in Singapore)
WABI: *Workshop on Algorithms in Bioinformatics* (2007 in Philadelphia, USA)

Example of national and European conferences:
ECCB: *European Conference on Computational Biology* (2008 in Cagliary, Italy)
GCB: *German Conference on Bioinformatics* (2007 in Potsdam)

Summary (Zusammenfassung):

1st Year (1. Jahr)	3.500,00 EUR
2nd Year (2. Jahr)	5.500,00 EUR
3rd Year (3. Jahr)	2.000,00 EUR
Sum (Summe) 4.4	**11.000,00 EUR**

4.5 Publication expenses (Publikationskosten)

The applicant declares the goal of concentrating the selection of his publication media on *open access*, which frequently incurs in publication fees. Nevertheless, no means for publication costs are requested for this project since the library of Bielefeld University supports this procedure through corporate memberships of *open-access* publishers such as BioMed Central.

Summary (Zusammenfassung):

1st Year (1. Jahr)	0,00 EUR
2nd Year (2. Jahr)	0,00 EUR
3rd Year (3. Jahr)	0,00 EUR
Sum (Summe) 4.5	**0,00 EUR**

4.6 Other costs (Sonstige Kosten)

None.

Summary (Zusammenfassung):

1st Year (1. Jahr)	0,00 EUR
2nd Year (2. Jahr)	0,00 EUR
3rd Year (3. Jahr)	0,00 EUR
Sum (Summe) 4.6	**0,00 EUR**

5 Prerequisites for carrying out the project (Voraussetzungen für die Durchführung des Vorhabens)

5.1 Your team (Zusammensetzung der Arbeitsgruppe)

The research group Genome Informatics was founded in March 2002, with financial support from the DFG Initiative “Bioinformatics”. As of now, it forms one of the central groups of the Bielefeld Institute for Bioinformatics. Two junior research groups are part of the group: *Computational Methods for Emerging Technologies* (Dr. Sven Rahmann; granted by Bielefeld University), and *Combinatorial Search Algorithms in Bioinformatics* (Dr. Ferdinando Cicalese; granted by the Sofja-Kovalevskaja-Prize of the Alexander-von-Humboldt Foundation). A former junior research group, *Computer Science Methods for Mass Spectrometry* (Prof. Dr. Sebastian Böcker; granted by the DFG-Action Plan Informatics/Emmy-Noether-Program), was dissolved as the group leader was promoted to professor at the University of Jena. The students of this group joined the other groups to continue their works.

5.2 Cooperation with other scientists (Zusammenarbeit mit anderen Wissenschaftlerinnen und Wissenschaftlern)

The applicant has several ongoing cooperations in diverse research areas, with the following ones being in direct relation to this application:

Prof. Stefan Kurtz, University of Hamburg and Prof. Robert Giegerich, Bielefeld University: cooperation on engineering of algorithms for efficient construction of suffix arrays.

Dr. Gene Myers, Howard Hughes Medical Institute/Janelia Farm: cooperation on engineering of algorithms for index-based, non-heuristic, approximate text search.

Prof. Bernd Weisshaar, chair of Genome Research, Bielefeld University: cooperation on large-scale repeat analysis with the goal of better primer design for EST detection in the genome of the sugar beet *Beta vulgaris*.

We also plan to start a cooperation with Dr. Felipe Rodrigues da Silva from Embrapa, the Brazilian National Agricultural Research Institution. Being involved in several genome projects in Brazil at present and in the past, Dr. da Silva will benefit from new algorithmic methods to be developed in this project to deal with the large amounts of data that will be produced in the near future. He will also contribute to the project with his expertise in genome projects and his extensive experience in genome assembly and annotation.

5.3 Foreign contacts and collaborations (Arbeiten im Ausland und Kooperation mit Partnern im Ausland)

As mentioned in the previous item, we plan to cooperate with Dr. Felipe Rodrigues da Silva from Embrapa, the Brazilian National Agricultural Research Institution, and this cooperation may involve visits from researchers of both sides.

5.4 Scientific equipment available (Apparative Ausstattung)

In early 2003, the Genome Informatics research group acquired workstations and servers from Sun Microsystems that were integrated in the CeBiTec infrastructure provided by the Bioinformatics Resource Facility (BRF), where they are available for use by any CeBiTec member. As a result, the members of the Genome Informatics group have complete access to the BRF infrastructure, including a cluster with more than 500 CPUs, a 10-Gigabit Ethernet network, a backup server with 54 Terabyte disc and 280 Terabytes of tape space, and various compute servers with up to 96 GB of main storage. This equipment is regularly being updated, most recently by an application of the CeBiTec in the HBFG program in summer 2007. Nevertheless, the need for computation with main memory demands above 32 GB as in the present application is rather unique in the CeBiTec environment. The only such computer presently available was bought by the applicant in 2003 and will soon run out of warranty. That is why we are applying here for the compute server mentioned in Section 4.2. Apart from that, according to statement of the BRF director, the existing infrastructure is sufficient for the requirements of the project proposed here. Personal computers for the scientific staff and students are available from the Genome Informatics research group resources.

5.5 Your institution's general contribution (Laufende Mittel für Sachausgaben)

The Genome Informatics group has the necessary infrastructure of a scientific working group with office rooms, office equipment and a secretary. The project requested here will benefit from this infrastructure made available at a value, roughly estimated, of 1,000 EUR annually (including office material and travel expenses).

5.6 Conflicts of interest with economic activities (Interessenkonflikte bei wirtschaftlichen Aktivitäten)

None.

5.7 Other requirements (Sonstige Voraussetzungen)

None.

6 Declaration (Erklärungen)

A request for funding this project has not been submitted to any other addressee. In the event that I submit such a request, I will inform the Deutsche Forschungsgemeinschaft immediately.

The DFG liaison officer of Bielefeld University, Herr Prof. Egelhaaf, was informed about this application.

7 Signature (Unterschrift)

Bielefeld, 19. August 2008

(Prof. Jens Stoye)

8 List of attachments (Verzeichnis der Anlagen)

1. Curriculum vitae of the applicant (DFG-Vordruck 10.04)
2. Complete publication list of the applicant
3. Letter of co-operation agreement from Dr. Felipe Rodrigues da Silva, Embrapa, Brazil
4. Letter of co-operation agreement from Prof. Bernd Weisshaar, Bielefeld University
5. Angebot for a computer from Moorbek Computer Systeme GmbH
6. CD-ROM with electronic versions of all request documents

All attachments do not need to be returned.

A References (Literatur zum Antrag)

[1] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *J. Discr. Alg.*, 2:53–86, 2004.

[2] A. Bergeron, S. Heber, and J. Stoye. Common intervals and sorting by reversals: A marriage of necessity. *Bioinformatics*, 18(Suppl. 2):S54–S63, 2002. (Proceedings of ECCB 2002).

[3] A. Bergeron, J. Mixtacki, and J. Stoye. Reversal distance without hurdles and fortresses. In *Proc. 15th Annual Symposium on Combinatorial Pattern Matching (CPM 2004)*, volume 3109 of *LNCS*, pages 388–399, 2004.

[4] A. Bergeron, J. Mixtacki, and J. Stoye. The inversion distance problem. In O. Gascuel, editor, *Mathematics of Evolution and Phylogeny*, pages 262–290, Oxford, UK, 2005. Oxford University Press.

[5] A. Bergeron, J. Mixtacki, and J. Stoye. On sorting by translocations. In S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. Waterman, editors, *Proceedings of the 9th Annual International Conference on Computational Molecular Biology, RECOMB*, volume 3500 of *LNCS*, pages 615–629, Berlin, 2005. Springer Verlag.

[6] A. Bergeron and J. Stoye. On the similarity of sets of permutations and its applications to genome comparison. In T. Warnow and B. Zhu, editors, *Proceedings of the 9th Annual International Conference on Computing and Combinatorics, COCOON 2003*, volume 2697 of *LNCS*, pages 68–79. Springer Verlag, 2003.

[7] S. A. de Carvalho Jr. and S. Rahmann. Improving the layout of oligonucleotide microarrays: Pivot Partitioning. In P. Bucher et al., editors, *Proceedings of the 6th Workshop of Algorithms in Bioinformatics*, volume 4175 of *Lecture Notes in Computer Science*, pages 321–332. Springer, 2006.

[8] S. A. de Carvalho Jr. and S. Rahmann. Improving the design of GeneChip arrays by combining placement and embedding. In *Proceedings of the 6th International Conference on Computational Systems Bioinformatics (CSB)*, 2007. To appear.

[9] R. Giegerich, S. Kurtz, and J. Stoye. Efficient implementation of lazy suffix trees. *Softw. Pract. Exper.*, 33(11):1035–1049, 2003.

[10] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, 1997.

[11] D. Gusfield and J. Stoye. Linear time algorithms for finding and representing all the tandem repeats in a string. *J. Comput. Syst. Sci.*, 69(4):525–546, 2004.

[12] S. Hannehalli. Polynomial-time algorithm for computing translocation distance between genomes. *Discrete Appl. Math.*, 71(1–3):137–151, 1996.

[13] S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Conf. Proc. 27th Annu. ACM Symp. Theory Comput., STOC 1995*, pages 178–189. ACM Press, 1995.

[14] S. Hannenhalli and P. A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proc. 36th Annu. Symp. Found. Comput. Sci., FOCS 1995*, pages 581–592. IEEE Press, 1995.

[15] S. Heber and J. Stoye. Algorithms for finding gene clusters. In O. Gascuel and B. Moret, editors, *Proceedings of the 1st Workshop on Algorithms in BioInformatics, WABI 01*, volume 2149 of *LNCS*, pages 254–265. Springer Verlag, 2001.

[16] S. Heber and J. Stoye. Finding all common intervals of k permutations. In A. Amir and G. Landau, editors, *Proceedings of the 12th Annual Symposium on Combinatorial Pattern Matching, CPM 01*, volume 2089 of *LNCS*, pages 207–218. Springer Verlag, 2001.

[17] R. M. Idury and M. S. Waterman. A new algorithm for DNA sequence assembly. *Journal of Computational Biology*, 2(2):291–306, 1995.

[18] S. Kurtz, J. V. Choudhuri, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich. REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res.*, 29(22):4643–4653, 2001.

[19] M. G. Maaß. Linear bidirectional on-line construction of affix trees. *Algorithmica*, 37(1):43–74, 2003.

[20] E. Moreno. *De Bruijn graphs and sequences in languages with restrictions*. PhD thesis, Université de Marne-La-Vallée, May 2005.

[21] E. W. Myers. The fragment assembly string graphs. *Bioinformatics*, 21:ii79–ii85, 2005.

[22] P. A. Pevzner, H. Tang, and G. Tesler. *De novo* repeat classification and fragment assembly. In *RECOMB'04*, pages 213–222, March 2004.

[23] P. A. Pevzner, H. Tang, and M. S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, August 2001.

[24] S. Rahmann. Fast large scale oligonucleotide selection using the longest common factor approach. *Journal of Bioinformatics and Computational Biology*, 1(2):343–361, 2003.

[25] S. Rahmann. The shortest common supersequence problem in a microarray production setting. *Bioinformatics*, 19(Suppl.2):ii156–ii161, 2003. ECCB special issue.

[26] S. Rahmann and C. Gräfe. Mean and variance of the Gibbs free energy of oligonucleotides in the nearest neighbor model under varying conditions. *Bioinformatics*, 20(17):2928–2933, 2004.

[27] B. Raphael, D. Zhi, H. Tang, and P. A. Pevzner. A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Research*, 14:2336–2346, 2004.

[28] K. Rasmussen, J. Stoye, and E. W. Myers. Efficient q-gram filters for finding all ϵ-matches over a given length. *J. Comp. Biol.*, 13(2):296–308, 2006.

[29] T. Schmidt and J. Stoye. Quadratic time algorithms for finding common intervals in two and more sequences. In *Proc. 15th Annual Symposium on Combinatorial Pattern Matching (CPM 2004)*, volume 3109 of *LNCS*, pages 347–358, 2004.

[30] K.-B. Schürmann and J. Stoye. An incomplex algorithm for fast suffix array construction. *Software: Practice and Experience*, 37(3):309–329, 2007.

[31] J. Stoye. Affix trees. Report 2000-04, Technische Fakultät der Universität Bielefeld, Abteilung Informationstechnik, 2000.

[32] J. Stoye and D. Gusfield. Simple and flexible detection of contiguous repeats using a suffix tree. *Theor. Comput. Sci.*, 270(1-2):843–856, 2002.