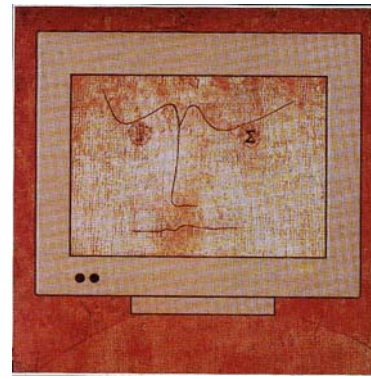
GWDG-Bericht Nr. 72

Kurt Kremer, Volker Macho (Hrsg.)

Forschung und wissenschaftliches Rechnen

Beiträge zum Heinz-Billing-Preis 2006



Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Forschung und
wissenschaftliches Rechnen

Titelbild:
Logo nach Paul Klee „Der Gelehrte“, Modifizierung durch I. Tarim,
Max-Planck-Institut für Psycholinguistik, Nijmegen.

GWDG-Bericht Nr. 72

Kurt Kremer, Volker Macho (Hrsg.)

Forschung und wissenschaftliches Rechnen

Beiträge zum Heinz-Billing-Preis 2006

Gesellschaft für wissenschaftliche Datenverarbeitung
Göttingen

2007

Die Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen ist eine gemeinsame Einrichtung der Max-Planck-Gesellschaft zur Förderung der Wissenschaften e. V. und des Landes Niedersachsen. Die Max-Planck-Gesellschaft hat diesen Bericht finanziell gefördert.

Redaktionelle Betreuung:
Volker Macho (Max-Planck-Institut fuer Polymerforschung, Mainz)

Satz:
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen,
Am Faßberg, D-37077 Göttingen

Druck: Artificium Stietenroth, D-37194 Bodenfelde, http://www.druckerei-stietenroth.de

ISSN: 0176-2516

Inhalt

Weitere Beiträge für den Heinz-Billing-Preis 2006

Vorwort

Dieser Band der Reihe „Forschung und wissenschaftliches Rechnen“ enthält sechs der sieben für den Heinz-Billing-Preis des Jahres 2006 eingereichten Beiträge.

Den Hauptpreis, welcher mit 3000,- € dotiert ist, erhielt Rafał Mantiuk vom Max-Planck-Institut für Informatik in Saarbrücken für seine Arbeit „High Dynamic Range Imaging: Towards the Limits of the Human Visual Perception“. High Dynamic Range Imaging (HDRI) ist ein Verfahren, welches es erlaubt, die bei Bildgebungsprozessen verwendete Farb- und Kontrastskala erheblich zu erweitern und somit die Fähigkeit des menschlichen Auges voll auszunutzen. Das Verfahren lässt eine Anwendung in vielen Bereichen der Wissenschaft und Technik erwarten, so dass man davon ausgehen kann, dass man hierüber in der Zukunft noch einiges hören wird.

Die weiteren Preise, welche mit jeweils 300,- € dotiert sind, erhielten Wolfgang Rieping aus dem Institut für Biochemie der Universität Cambridge zusammen mit Michael Habeck vom Max-Planck-Institut für Entwicklungsbiologie in Tübingen für die Arbeit „ISD - A Bayesian Software for Protein Structure Determination“ und Johannes Söding, ebenfalls vom Max-Planck-Institut für Entwicklungsbiologie, Tübingen, für die Arbeit „Protein Structure and Function Prediction by Pairwise Comparison of Hidden Markov Models“.

Im Jahre 2006 wurde die „Heinz-Billing-Stiftung“ der Max-Planck-Gesellschaft gegründet. Sie verwaltet das Vermögen der bisherigen Heinz-Billing-Vereinigung und wird von einem Stiftungsrat geleitet. Es ist die

Aufgabe des Stiftungsrates, den Preis auch in Zukunft auszuloben und den sich ändernden wissenschaftlichen Gegebenheiten anzupassen. Damit ist ein wesentlicher Schritt zur Weiterführung des Preises in den kommenden Jahren getan. Der Vorstand der erloschenen Heinz-Billing-Vereinigung möchte sich an dieser Stelle für die Unterstützung durch das Präsidium der MPG und der Generalverwaltung bedanken.

Mit dieser Entwicklung werden einige weitere Änderungen einhergehen. Dazu gehört auch die Form der Publikation der ausgezeichneten Arbeiten. Wir als Herausgeber möchten uns daher an dieser Stelle bei Ihnen als Lesern ganz herzlich verabschieden.

Unser Dank gilt ganz besonders wie in jedem Jahr Herrn Günter Koch von der Gesellschaft für wissenschaftliche Datenverarbeitung in Göttingen für die Gestaltung und die Erstellung einer druckreifen Vorlage.

Die Vergabe des Preises wäre ohne Sponsoren nicht möglich. Wir danken der Firma IBM Deutschland, welche wiederholt als Hauptsponsor aufgetreten ist, für ihre großzügige Unterstützung.

Die hier abgedruckten Arbeiten sind ebenfalls im Internet unter der Adresse

www. billingpreis.mpg.de

zu finden.

Kurt Kremer, Volker Macho

Der Heinz-Billing-Preis 2006

Ausschreibung des Heinz-Billing-Preises 2006 zur Förderung des wissenschaftlichen Rechnens

Im Jahre 1993 wurde zum ersten Mal der Heinz-Billing-Preis zur Förderung des wissenschaftlichen Rechnens vergeben. Mit dem Preis sollen die Leistungen derjenigen anerkannt werden, die in zeitintensiver und kreativer Arbeit die notwendige Hard- und Software entwickeln, die heute für neue Vorstöße in der Wissenschaft unverzichtbar sind.

Der Preis ist benannt nach Professor Heinz Billing, emiritiertes wissenschaftliches Mitglied des Max-Planck-Institutes für Astrophysik und langjähriger Vorsitzender des Beratenden Ausschusses für Rechenanlagen in der Max-Planck-Gesellschaft. Professor Billing stand mit der Erfindung des Trommelspeichers und dem Bau der Rechner G1, G2, G3 als Pionier der elektronischen Datenverarbeitung am Beginn des wissenschaftlichen Rechnens.

Der Heinz-Billing-Preis steht unter dem Leitmotiv

"EDV als Werkzeug der Wissenschaft".

Für den Heinz-Billing-Preis können Arbeiten eingereicht werden, die in besonderer Weise Beispiele dafür sind, wie die EDV als methodisches Werkzeug Forschungsgebiete unterstützt oder einen neuen Forschungsansatz ermöglicht hat. Folgender Stichwortkatalog mag als Anstoß dienen:

- Implementation von Algorithmen und Softwarebibliotheken
- Modellbildung und Computersimulation
- Gestaltung des Benutzerinterfaces
- EDV gestützte Messverfahren
- Datenanalyse und Auswertungsverfahren
- Visualisierung von Daten und Prozessen

Die eingereichten Arbeiten werden referiert und in der Buchreihe "Forschung und wissenschaftliches Rechnen" veröffentlicht.

Die Jury wählt einen Beitrag für den mit insgesamt 3000,- dotierten Heinz-Billing-Preis 2006 zur Förderung des wissenschaftlichen Rechnens aus. Für die Beiträge auf den Plätzen 2 und 3 werden jeweils 300,- Euro vergeben. Die Beiträge zum Heinz-Billing-Preis, in deutscher oder englischer Sprache abgefasst, müssen keine Originalarbeiten sein und sollten möglichst nicht mehr als fünfzehn Seiten umfassen.

Da zur Bewertung eines Beitrags im Sinne des Heinz-Billing-Preises neben der technischen EDV-Lösung insbesondere der Nutzen für das jeweilige Forschungsgebiet herangezogen wird, sollte einer bereits publizierten Arbeit eine kurze Ausführung zu diesem Aspekt beigefügt werden

Bis zum Jahre 2005 wurde der Preis von der Heinz-Billing-Vereinigung zur Förderung des wissenschaftlichen Rechnens e.V. vergeben. Ab 2006 wird die Vergabe vom Stiftungsrat der Heinz-Billing-Stiftung der Max-Planck-Gesellschaft vorgenommen

Beiträge für den Heinz-Billing-Preis sind bis zum 30. Juni 2006 an folgende Adresse einzureichen:

Prof. Dr. Kurt Kremer
Max-Planck-Institut für Polymerforsch
Ackermannweg 10, 55128 Mainz
Tel.: 06131/379-140
Fax: 06131/379-430
e-mail:kremer@mpip-mainz.mpg.de

Heinz-Billing-Preisträger

1993: Dr. Hans Thomas Janka, Dr. Ewald Müller, Dr. Maximilian Ruffert
Max-Planck-Institut für Astrophysik, Garching
Simulation turbulenter Konvektion in Supernova-Explosionen in massereichen Sternen

1994: Dr. Rainer Goebel
Max-Planck-Institut für Hirnforschung, Frankfurt
Neurolator - Ein Programm zur Simulation neuronaler Netzwerke

1995: Dr. Ralf Giering
Max-Planck-Institut für Meteorologie, Hamburg
AMC: Ein Programm zum automatischen Differenzieren von Fortran Programmen

1996: Dr. Klaus Heumann
Max-Planck-Institut für Biochemie AG MPIS, Martinsried
Systematische Analyse und Visualisierung kompletter Genome am Beispiel von S. cerevisiae

1997: Dr. Florian Mueller
Max-Planck-Institut für molekulare Genetik, Berlin
ERNA-3D (Editor für RNA - Dreidimensional) (PDF-Format)

1998: Prof. Dr. Edward Seidel
Max-Planck-Institut für Gravitationsphysik Albert-Einstein-Institut, Potsdam
Technologies for Collaborative, Large Scale Simulation in Astrophysics and a General Toolkit for solving PDEs in Science and Engineering (PDF-Format)

1999: Dr. Alexander Pukhov
Max-Planck-Institut für Quantenoptik, Garching
Three-dimensional relativistic electromagnetic Particle-in-Cell code VLPL - Virtual Laser Plasma Laboratory (PDF-Format)

2000: Dr. Oliver Kohlbacher
Max-Planck-Institut für Informatik, Saarbrücken
BALL - A Framework for Rapid Application Development in Molecular Modeling (PDF-Format)

2001: Dr. Jörg Haber
Max-Planck-Institut für Informatik, Saarbrücken
MEDUSA, ein Software-System zur Modellierung und Animation von Gesichtern

2002: Daan Broeder, Hennie Brugman und Reiner Dirksmeyer
Max-Planck-Institut für Psycholinguistik, Nijmegen
NILE - Nijmegen Language Resource Environment

2003: Roland Chrobok, Sigurður F. Hafstein und Andreas Pottmeier
Universität Duisburg-Essen
OLSIM: A New Generation of Traffic Information Systems

2004: Markus Rampp, Thomas Soddemann
Rechenzentrum Garching der Max-Planck-Gesellschaft, Garching
A Work Flow Engine for Microbial Genome Research

2005: Patrick Jöckel, Rolf Sander
Max-Planck-Institut für Chemie, Mainz
The Modular Earth Submodel System (MESSy)

2006: Rafał Mantiuk
High Dynamic Range Imaging: Towards the Limits of the Human Visual Perception

Rafał Mantiuk,
Max-Planck-Institut für Informatik, Saarbrücken

erhält den

Heinz-Billing-Preis 2006

zur Förderung

des wissenschaftlichen Rechnens

als Anerkennung für seine Arbeit

High Dynamic Range Imaging:
Towards the Limits of the Human Visual Perception

Laudatio

Die heute übliche Speicher- und Aufnahmetechnik erlaubt es, nur einen Bruchteil der Farbskala, die vom menschlichen Auge erfasst wird, zu verarbeiten. Im Rahmen des High Dynamic Range Imaging (HDRI) will man die Farb- und Kontrastskala wesentlich erweitern. Das ist mit ersten HDRI-Kameras und Monitoren darstellbar. Herr Mantiuk hat mit seinem Programm „pftools“ ein effizientes und wichtiges Werkzeug für die Darstellung des Farbraums mit 32 Bit Fließkommawerten geschaffen und eine dem menschlichen Auge angepasste diskretisierte Komprimierung entwickelt. Das Verfahren lässt eine Anwendung in vielen Bereichen der Wissenschaft und Technik erwarten.

Nach der Überreichung der Urkunde: Prof. Kremer (links) mit dem Preisträger

High Dynamic Range Imaging: Towards the Limits of the Human Visual Perception

Rafał Mantiuk
Max-Planck-Institut für Informatik
Saarbrücken

1 Introduction

Vast majority of digital images and video material stored today can capture only a fraction of visual information visible to the human eye and does not offer sufficient quality to reproduce them on the future generation of display devices. The limitating factor is not the resolution, since most consumer level digital cameras can take images of higher number of pixels than most of displays can offer. The problem is a limited color gamut and even more limited dynamic range (contrast) that cameras can capture and that majority of image and video formats can store.

For instance, each pixel value in the JPEG image encoding is represented using three 8-bit integer numbers (0-255) using the YC_rC_b color space. Such color space is able to store only a small part of visible color gamut (although containing the colors most often encountered in the real world), as illustrated in Figure 1-left, and even smaller part of luminance range that can be perceived by our eyes, as illustrated in Figure 1-right. The reason for this is that the JPEG format was designed to store as much information as can be displayed on the majority of displays, which were Cathode Ray Tube (CRT)

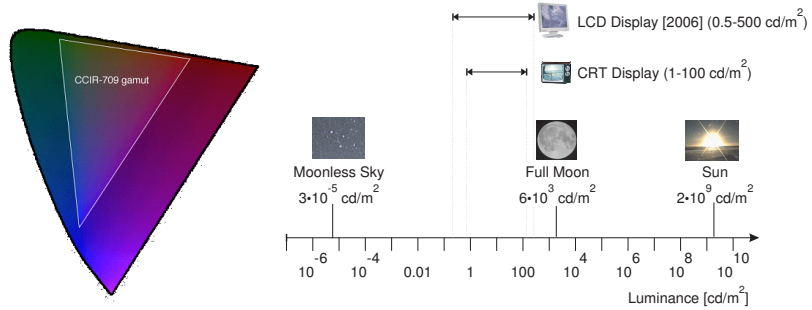


Fig. 1: Left: color gamut frequently used in traditional imaging (CCIR-705), compared to the full visible color gamut. Right: real-world luminance values compared with the range of luminance that can be displayed on CRT and LDR monitors.

monitors at the time when the JPEG compression was developed. This assumption is no longer valid, as the new generations of LCD and Plasma displays can visualize much broader color gamut and dynamic range than their CRT ancestors. Moreover, as new display devices become available, there is a need for higher precision of image and video content. The traditional low-dynamic range and limited color gamut imaging, which is confined to three 8-bit integer color channels, cannot offer the precision that is needed for the further developments in image capture and display technologies.

The High Dynamic Range Imaging (HDRI) overcomes the limitation of traditional imaging by using much higher precision when performing operations on color. Pixel colors are specified in HDR images as a triple of floating point values (usually 32-bit per color channel), providing the accuracy that is far below the visibility threshold of the human eye. Moreover, HDRI operates on colors of original scenes, instead of their renderings on a particular display medium, as is the case of the traditional imaging. By its inherent colorimetric precision, HDRI can represent all colors that can be found in real world and can be perceived by the human eye.

HDRI, which originated from the computer graphics field, has been recently gaining momentum and revolutionizing almost all fields of digital imaging. One of the breakthroughs of the HDR revolution was the development of an HDR display, which proved that the visualization of color and the luminance range close to real scenes is possible (Seetzen, Heidrich, Stuerzlinger, Ward, Whitehead, Trentacoste, Ghosh & Vorozcovs 2004). One of the first to adopt HDRI were video game developers together with graphics card vendors. Today most of the state-of-the art video game engines perform rendering using HDR precision to deliver more believable and appealing virtual reality worlds. A computer generated imagery used in the special

effect production strongly depends on the HDR techniques. High-end cinematographic cameras, both analog and digital, already provide significantly higher dynamic range than most of the displays today. Their quality can be retained after digitalization only if a form of HDR representation is used. HDRI is also a strong trend in digital photography, mostly due to the multi-exposure techniques, which can be used to take an HDR image even with a consumer level digital camera. To catch up with the HDR trend, many software vendors announce their support of the HDR image formats, taking Adobe® Photoshop® CS2 as an example.

Besides its significant impact on existing imaging technologies that we can observe today, HDRI has potential to radically change the methods in which imaging data is processed, displayed and preserved in several fields of science. Computer vision algorithms can greatly benefit from the increased precision of HDR images, which lack over- or under-exposed regions, which are often the cause of the algorithms failure. Medical imaging has already developed image formats (DICOM format) that can partly cope with shortcomings of traditional images, however they are supported only by specialized hardware and software. HDRI gives the sufficient precision for medical imaging and therefore its capture, processing and rendering techniques can be used also in this field. For instance, HDR displays can show even better contrast than high-end medical displays and therefore facilitate diagnosing based on CT scans. HDR techniques can also find applications in astronomical imaging, remote sensing, industrial design and scientific visualization.

HDRI does not only provide higher precision, but also enables to synthesize, store and visualize a range of perceptual cues, which are not achievable with the traditional imaging. Most of the imaging standards and color spaces have been developed to match the needs of office or display illumination conditions. When viewing such scenes or images in such conditions, our visual system operates in a mixture of day-light and dim-light vision state, so called the mesopic vision. When viewing out-door scenes, we use day-light perception of colors, so called the photopic vision. This distinction is important for digital imaging as both types of vision shows different performance and result in different perception of colors. HDRI can represent images of luminance range fully covering both the photopic and the mesopic vision, thus making distinction between them possible. One of the differences between mesopic and photopic vision is the impression of colorfulness of objects. We tend to regard objects more colorful when they are brightly illuminated, which is the phenomena that is called Hunt's effect. To render enhanced colorfulness properly, digital images must preserve information about the actual level of luminance of the original scene, which is not possible in the case of the traditional imaging. Real-world scenes are not only brighter and more colorful than their digital reproductions, but also contain much higher contrast, both

local between neighboring objects, and global between distant objects. The eye has evolved to cope with such high contrast and its presence in a scene evokes important perceptual cues. The traditional imaging, unlike HDRI, is not able to represent such high-contrast scenes. Similarly, the traditional images can hardly represent such common visual phenomena as self-luminous surfaces (sun, shining lamps) and bright specular highlights. They also do not contain enough information to reproduce visual glare (brightening of the areas surrounding shining objects) and a short-time dazzle due to sudden raise of light level (e.g. when exposed to the sunlight after staying indoors). To faithfully represent, store and then reproduce all these effects, the original scene must be stored and treated using high fidelity HDR techniques.

Despite its advantages, the inception of HDRI in various fields of digital imaging poses serious problems. The biggest is the lack of well standardized color spaces and image formats, of which traditional imaging is abundant. Such color spaces and image formats would facilitate exchange of information between HDR applications. Due to the different treatment of color, introduction of HDRI also requires redesigning entire imaging pipeline, including acquisition (cameras, computer graphics synthesis), storage (formats, compression algorithms) and display (HDR display devices and display algorithms).

This paper summarizes the work we have done to make the transition from the traditional imaging to HDRI smoother. In the next section we describe our implementation of HDR image and video processing framework, which we created for the purpose of our research projects and which we made available as an Open Source project. Section 3 describes our contributions in the field of HDR image and video encoding. These include a perceptually motivated color space for efficient encoding of HDR pixels and two extensions of MPEG standard that allow to store movies containing full color gamut and luminance range visible to the human eye.

2 HDR Imaging Framework

Most of the traditional image processing libraries store each pixel using limited-precision integer numbers. Moreover, they offer restricted means of colorimetric calibration. To overcome these problems, we have implemented HDR imaging framework as a package of several command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. The package was intended to solve our current research problems, therefore simplicity and flexibility were priorities in its design. Since we found the software very useful in numerous projects, we decided to make it available for the research community as an Open Source project

licensed under the GPL. The software is distributed under the name *pfstools* and its home page can be found at `http://pfstools.sourceforge.net/`.

The major role of the software is the integration of several imaging and image format libraries, such as *ImageMagick*, *OpenEXR* and *NetPBM*, into a single framework for processing high precision images. To provide enough flexibility for a broad range of applications, we have build *pfstools* on the following concepts:

- Images/frames should hold an arbitrary number of channels (layers), which can represent not only color, but also depth, alpha-channel, and texture attributes;
- Each channel should be stored with high precision, using floating point numbers. If possible, the data should be colorimetrically calibrated and provide the precision that exceeds the performance of the human visual system.
- Luminance should be stored using physical units of cd/m^2 to distinguish between the night- and the day-light vision.
- There should be user-defined data entries for storing additional, application specific information (e.g. colorimetric coordinates of the white point).

pfstools are built around a generic and simple format of storing images, which requires only a few lines of code to read or write. The format offers arbitrary number of channels, each represented as a 2-D array of 32-bit floating point numbers. There is no compression as the files in this format are intended to be transferred internally between applications without writing them to a disk. A few channels have a predefined function. For example, channels with the IDs 'X', 'Y' and 'Z' are used to store color data in the CIE XYZ (absolute) color space. This is different to most imaging frameworks that operate on RGB channels. The advantage of the CIE XYZ color space is that it is precisely defined in terms of spectral radiance and the full visible color gamut can be represented using only positive values of color components. The file format also offers a way to include in an image any number of user *tags* (name and value pairs), which can contain any application dependent data. A sequence of images is interpreted by all "pfs-compliant" applications as consequtive frames of an animation, so that video can be processed in the same way as images. The format is described in detail in a separate specification[1].

pfstools are a set of command line tools with almost no graphical user interface. This greatly facilitates scripting and lessens the amount of work needed to program and maintain a user interface. The exception is a viewer

[1] Specification of the *pfs* format can be found at:
`http://www.mpi-sb.mpg.de/resources/pfstools/pfs_format_spec.pdf`

of HDR images. The main components of *pfstools* are: programs for reading and writing images in all major HDR and LDR formats (e.g. OpenEXR, Radiance's RGBE, logLuv TIFF, 16-bit TIFF, PFM, JPEG, PNG, etc.), programs for basic image manipulation (rotation, scaling, cropping, etc.), an HDR image viewer, and a library that simplifies file format reading and writing in C++. The package includes also an interface for *matlab* and *GNU Octave*. The *pfstools* framework does not impose any restrictions on the programming language. All programs that exchange data with *pfstools* must read or write the file format, but there is no need to use any particular library. The typical usage of *pfstools* involves executing several programs joined by UNIX pipes. The first program transmits the current frame or image to the next one in the chain. The final program should either display an image or write it to a disk. Such pipeline architecture improves flexibility of the software but also gives straightforward means for parallel execution of the pipeline components on multiprocessor computers. Some examples of command lines are given below:

```
pfsin␣input.exr␣|␣pfsfilter␣|␣pfsout␣output.exr
```

Read the image `input.exr`, apply the filter `pfsfilter` and write the output to `output.exr`.

```
pfsin␣input.exr␣|␣pfsfilter␣|␣pfsview
```

Read the image `input.exr`, apply the filter `pfsfilter` and show the result in an HDR image viewer.

```
pfsin␣in%04d.exr␣--frames␣100:2:200␣\
␣|␣pfsfilter␣|␣pfsout␣out%04d.hdr
```

Read the sequence of OpenEXR frames `in0100.exr`, `in0102.exr`, .., `in0200.exr`, apply the filter `pfsfilter` and write the result in Radiance's RGBE format to `out0000.hdr`, `out0001.hdr`, ...

---

pfstools is only a base set of tools which can be easily extended and integrated with other software. For example, *pfstools* is used to read, write and convert images and video frames for the prototype implementation of our image and video compression algorithms. HDR images can be rendered on existing displays using one of the several implemented tone mapping algorithms from the *pfstmo* package[2], which is build on top of *pfstools*. Using the software from the *pfscalibration* package[3], which is also based on *pfstools*,

[2] *pfstmo* home page: `http://www.mpii.mpg.de/resources/tmo/`

[3] *pfscalibration* home page: `http://www.mpii.mpg.de/resources/hdr/calibration/pfs.html`

cameras can be calibrated and images rescaled in physical or colorimetrical units. A computational model of the human visual system – *HDR-VDP*[4] – uses *pfstools* to read its input from multitude of image formats.

We created *pfstools* to fill the gap in the imaging software, which can seldom handle HDR images. We have found from the e-mails we received and the discussion group contacts that *pfstools* is used for high definition HDR video encoding, medical imaging, variety of tone mapping projects, texture manipulations and quality evaluation of CG rendering.

3 HDR Image and Video Compression

Wide acceptance of new imaging technology is hardly possible if there is no image and video content that the users could benefit from. The distribution of digital content is strongly limited if there is no efficient image and video compression and no standard file formats that software and hardware could recognize and read. In this section we propose several solutions to the problem of HDR image and video compression, including a color space for HDR pixels that is used as an extension to the MPEG-4 standard, and a backward-compatible HDR MPEG compression algorithm.

3.1 Color Space for HDR Pixels

Although the most natural representation of HDR images is a triple of floating point numbers, such representation does not lead to the best image or video compression ratios and adds complexity to compression algorithms. Moreover, since the existing image and video formats, such as MPEG-4 or JPEG2000, can encode only integer numbers, HDR pixels must be represented as integers in order to encode them using these formats. Therefore, it is highly desirable to convert HDR pixels from a triple of 32-bit floating point values, to integer numbers. Such integer encoding of luminance should take into account the limitations of human perception and the fact that the eye can see only limited numbers of luminance levels and colors. This section gives an overview of the color space that can efficiently represent HDR pixel values using only integer numbers and the minimal number of bits. More information on this color space can be found in (Mantiuk, Myszkowski & Seidel 2006).

Different applications may require different precision of the visual data. For example satellite imaging may require multi-spectral techniques to capture information that is not even visible to the human eye. However, for a

[4] *HDR-VDP* home page: `http://www.mpii.mpg.de/resources/hdr/vdp/index.html`

large number of applications it is sufficient if the human eye cannot notice any encoding artifacts. It is important to note that low dynamic range formats, like JPEG or a simple profile MPEG, can not represent the full range of colors that the eye can see. Although the quantization artifacts due to 8-bit discretization in those formats are hardly visible to our eyes, those encoding can represent only the fraction of the dynamic range and the color gamut that the eye can see.

Choice of the color space used for image or video compression has a great impact on the compression performance and the capabilities of the encoding format. To offer the best trade-off between compression efficiency and visual quality without imposing any assumptions on the display technology, we propose that the color space used for compression has the following properties:

1. The color space can encode the full color gamut and the full range of luminance that is visible to the human eye. This way the human eye, instead of the current imaging technology, defines the limits of such encoding.
2. A unit distance in the color space correlates with the Just Noticeable Difference (JND). This offers a more uniform distribution of distortions across an image and simplifies control over distortions for lossy compression algorithms.
3. Only positive integer values are used to encode luminance and color. Integer representation simplifies and improves image and video compression.
4. A half-unit distance in the color space is below 1 JND. If this condition is met, the quantization errors due to rounding to integer numbers are not visible.
5. The correlation between color channels should be minimal. If color channels are correlated, the same information is encoded twice, which worsens the compression performance.
6. There is a direct relation between the encoded integer values and the photometrically calibrated XYZ color values.

There are several color spaces that already meet some of the above requirements, but there is no color space that accommodates them all. For example, the Euclidean distance in the *CIE* $L^*u^*v^*$ color space correlates with the JND (Property 2), but this color space does not generalize to the full range of visible luminance levels, ranging from scotopic light levels, to very bright photopic conditions. Several perceptually uniform quantization strategies have been proposed (Sezan, Yip & Daly 1987, Lubin & Pica 1991), including the grayscale standard display function from the DICOM standard (DICOM PS 3-2004 2004). However, none of these take into account as broad dynamic range and diversified luminance conditions as required by Property 1.

Most of the traditional image or video formats use so called *gamma correction* to convert luminance or RGB tristimulus values into integer numbers, which can be latter encoded. Gamma correction is usually given in a form of

a power function $intensity = signal^{\gamma}$ (or $signal = intensity^{(1/\gamma)}$ for an inverse gamma correction), where the value of γ is typically around 2.2. Gamma correction was originally intended to reduce camera noise and to control the current of the electron beam in CRT monitors. Further details on gamma correction can be found in (Poynton 2003). Accidentally, light *intensity* values, after being converted into *signal* using the inverse gamma correction formula, correspond usually well with our perception of lightness. Therefore such values are also well suited for image encoding since the distortions caused by image compression are equally distributed across the whole scale of *signal* values. In other words, altering *signal* by the same amount for both small values and large values of a signal should result in the same magnitude of visible changes. Unfortunately, this is only true for a limited range of luminance values, usually within a range from 0.1 to 100 cd/m^2. This is because the response characteristics of the human visual system (HVS) to luminance[5] changes considerably above 100 cd/m^2. This is especially noticeable for HDR images, which can span the luminance range from 10^{-5} to 10^{10} cd/m^2. An ordinary gamma correction is not sufficient in such case and a more elaborate model of luminance perception is needed. This problem is solved by the *JND* encoding, described in this section.

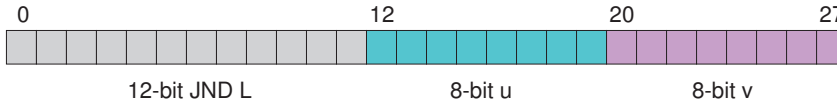


Fig. 2: 28-bit per pixel JND encoding

JND encoding can be regarded as an extension of gamma correction to HDR pixel values. The name *JND* encoding is motivated by its design, which makes the encoded values correspond to the Just Noticeable Differences (JND) of luminance.

JND encoding requires two bytes to represent color and 12 bits to encode luminance (see Figure 2). Chroma (hue and saturation) is represented using u' and v' chromacities as recommended by CIE 1976 Uniform Chromacity Scales (UCS) diagram and defined by equations:

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{1}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{2}$$

[5]HVS use both types of photoreceptors, cones and rods, in the range of luminance aproximately from 0.1 to 100 cd/m^2. Above 100 cd/m^2 only cones contribute to the visual response.

Luma, l, is found from absolute luminance values, y $[cd/m^2]$, using the formula:

$$l_{hdr}(y) = \begin{cases} a \cdot y & \text{if } y < y_l \\ b \cdot y^c + d & \text{if } y_l \leq y < y_h \\ e \cdot \log(y) + f & \text{if } y \geq y_h \end{cases} \quad (3)$$

There is also a formula for the inverse conversion, from 12-bit luma to luminance:

$$y(l_{hdr}) = \begin{cases} a' \cdot l_{hdr} & \text{if } l_{hdr} < l_l \\ b'(l_{hdr} + d')^{c'} & \text{if } l_l \leq l_{hdr} < l_h \\ e' \cdot \exp(f' \cdot l_{hdr}) & \text{if } l_{hdr} \geq l_h \end{cases} \quad (4)$$

The constants are given in the table below:

$a = 17.554$	$e = 209.16$	$a' = 0.056968$	$e' = 32.994$
$b = 826.81$	$f = -731.28$	$b' = 7.3014e-30$	$f' = 0.0047811$
$c = 0.10013$	$y_l = 5.6046$	$c' = 9.9872$	$l_l = 98.381$
$d = -884.17$	$y_h = 10469$	$d' = 884.17$	$l_h = 1204.7$

The above formulas have been derived from the psychophysical measurements of the luminance detection thresholds[6]. To meet our initial requirements for HDR color space, in particular Property 4, the derived formulas guarantee that the same difference of values l, regardless whether in bright or in dark region, corresponds to the same visible difference. Neither luminance nor the logarithm of luminance has this property, since the response of the human visual system to luminance is complex and non-linear. The values of l lay in the range from 0 to 4095 (12 bit integer) for the corresponding luminance values from 10^{-5} to 10^{10} cd/m^2, which is the range of luminance that the human eye can effectively see (although the values above 10^6 can be damaging to the eye and would mostly be useful for representing the luminance of bright light sources).

Function $l(y)$ (Equation 3) is plotted in Figure 3 and labelled "*JND* encoding". Note that both the formula and the shape of the *JND* encoding is very similar to the nonlinearity (gamma correction) used in the sRGB color space. Both *JND* encoding and sRGB nonlinearity follow similar curve on the plot, but the *JND* encoding is more conservative (a steeper curve means that a luminance range is projected on a larger number of discrete luma values, V, thus lowering quantization errors). However, the sRGB non-linearity results in a too steep function for luminance above 100 cd/m^2, which requires too many bits to encode real-world luminance values.

The color space described in this section can be directly used for many existing image and video compression formats, such as JPEG-2000 and MPEG-4.

[6]The full derivation of this function can be found in (Mantiuk, Myszkowski & Seidel 2006). The formulas are derived from the threshold versus intensity characteristic measured for human subjects and fitted to the analytical model (CIE 1981).

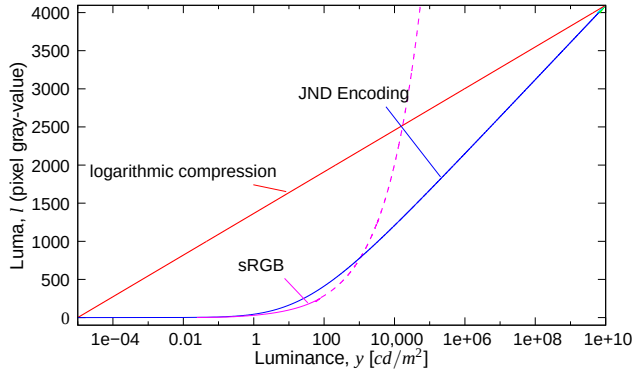


Fig. 3: Functions mapping physical luminance y to encoded luma values l. JND Encoding – perceptual encoding of luminance; sRGB – nonlinearity (gamma correction) used for the sRGB color space; logarithmic compression – logarithm of luminance, rescaled to 12-bit integer range. Note that encoding high luminance values using the sRGB nonlinearity (dashed line) would require significantly larger number of bits than the perceptual encoding.

Both these formats can encode luminance with 12 or more bits, which make them fully capable of representing HDR pixel values. As a proof of concept we extended an MPEG-4 compression algorithm to use the proposed color space. The modified video encoder achieved good compression performance, offering the ability to store the full color gamut and the range of luminance that is visible to the human eye (Mantiuk, Krawczyk, Myszkowski & Seidel 2004), as demonstrated in Figure 4. Moreover, the advanced HDR video player, which we created for the purpose of playback of HDR movies, can play video and apply one from several available tone-mapping algorithms in real-time (Krawczyk, Myszkowski & Seidel 2005). The additional advantage of HDR content is the possibility to simulate on traditional displays the perceptual effects that are normally only evoked when observing scenes of large contrast and luminance range. An examples of such effects are the night vision and an optically accurate motion blur, demonstrated in Figure 5. More examples can be found at the project page: `http://www.mpi-inf.mpg.de/resources/hdrvideo/index.html`.

The application of the proposed color space is not limited to image and video encoding. Since the color space is approximately perceptually uniform (Property 2), it can be used as a color difference metric for HDR images, similarly as the *CIE* $L^*u^*v^*$ color space is commonly used for traditional images. The luminance coding can also approximate photoreceptor response to light in the computational models of the human visual system (Mantiuk, Myszkowski & Seidel 2006). Since the proposed color encoding minimizes the number of bits required to represent color and at the same time does not

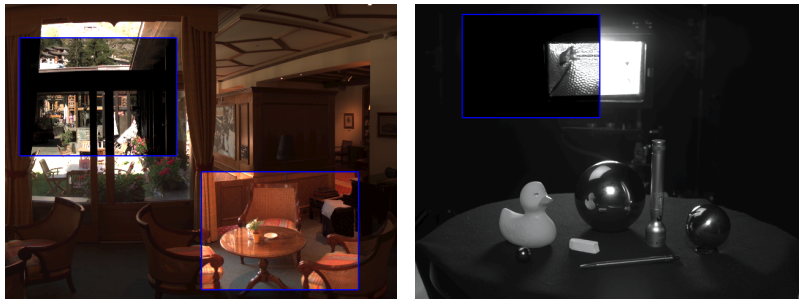


Fig. 4: Two screenshots from the advanced HDR video player, showing an extreme dynamic range captured withing HDR video sequences. Blue frames represent virtual filters that adjust exposure in the selected regions.

Fig. 5: Screenshots demonstrating simulation of perceptual and optical effects, possible only for HDR content. Left: simulation of night vision, resulting in a limited color vision and bluish cast of colors. Right: simulation of physically accurate motion blur (right side) compared with the motion blur computed from the traditional video material (left side).

compromise visual quality, it can be an attractive method of encoding data transmitted digitally from the CPU to a graphics card or from the graphics card to a display device.

3.2 Backward-compatible HDR Video Compression

Since the traditional, low-dynamic range (LDR) file formats for images and video, such as JPEG or MPEG, have become widely adapted standards, supported by almost all software and hardware equipment dealing with digital imaging, it cannot be expected that these formats will be immediately replaced with their HDR counterparts. To facilitate transition from the traditional to HDR imaging, there is a need for backward compatible HDR formats, that would be fully compatible with existing LDR formats and at the same time would support enhanced dynamic range and color gamut.

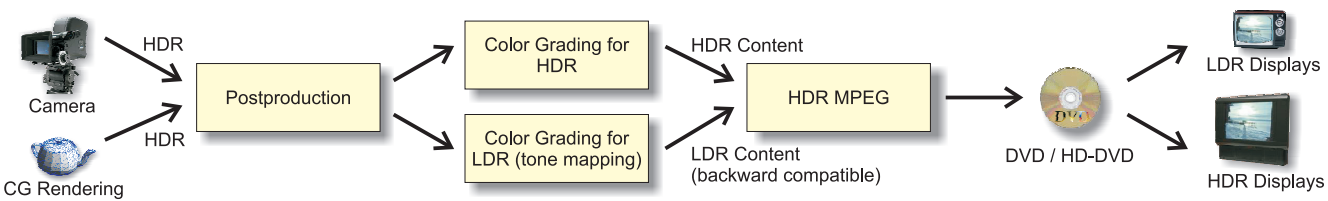


Fig. 6: The proposed backward compatible HDR DVD movie processing pipeline. The high dynamic range content, provided by advanced cameras and CG rendering, is encoded in addition to the low dynamic range (LDR) content in the video stream. The files compressed with the proposed HDR MPEG method can play on traditional LDR and future generation HDR displays.

Encoding movies in HDR format is attractive for cinematography, especially that movies are already shot with high-end cameras, both analog and digital, that can capture much higher dynamic range than typical MPEG compression can store. To encode cinema movies using traditional MPEG compression, the movie must undergo processing called color grading. Part of this process is the adjustment of tones (tone-mapping) and colors (gamut-mapping), so that they can be displayed on majority of TV sets (refer to Figure 6). Although such processing can produce high quality content for typical CRT and LCD displays, the high quality information, from which advanced HDR displays could benefit, is lost. To address this problem, the proposed HDR-MPEG encoding can compress both LDR and HDR into the same backward compatible movie file (see Figure 6). Depending on the capabilities of the display and playback hardware or software, either LDR or HDR content is displayed. This way HDR content can be added to the video stream at the moderate cost of about 30% of the LDR stream size. Because of such small overhead, both standard-definition and high-definition (HD) movies can fit in their original storage medium when encoded with HDR information.

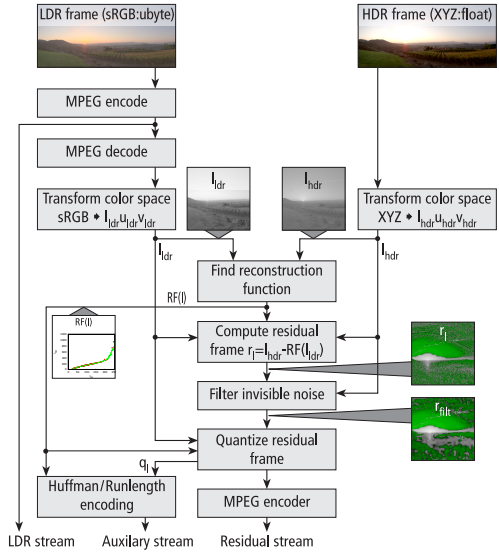


Fig. 7: A data flow of the backward compatible HDR MPEG encoding.

The complete data flow of the proposed backward compatible HDR video compression algorithm is shown in Figure 7. The encoder takes two sequences of HDR and LDR frames as input. The LDR frames, intended for LDR devices, usually contain a tone mapped or gamut mapped version of the

original HDR sequence. The LDR frames are compressed using a standard MPEG encoder (*MPEG encode* in Figure 7) to produce a backward compatible LDR stream. The LDR frames are then decoded to obtain a distorted (due to lossy compression) LDR sequence, which is later used as a reference for the HDR frames (see *MPEG decode* in Figure 7). Both the LDR and HDR frames are then converted to compatible color spaces, which minimize differences between LDR and HDR colors. The reconstruction function (see *Find reconstruction function* in Figure 7) reduces the correlation between LDR and HDR pixels by giving the best prediction of HDR pixels based on the values of LDR pixels. The residual frame is introduced to store a difference between the original HDR values and the values predicted by the reconstruction function. To further improve compression, invisible luminance and chrominance variations are removed from the residual frame (see *Filter invisible noise* in Figure 7). Such filtering simulates the visual processing that is performed by the retina in order to estimate the contrast detection threshold at which the eye does not see any differences. The contrast magnitudes that are below this threshold are set to zero. Finally, the pixel values of a residual frame are quantized (see *Quantize residual frame* in Figure 7) and compressed using a standard MPEG encoder into a residual stream. Both the reconstruction function and the quantization factors are compressed using a lossless arithmetic encoding and stored in an auxiliary stream.

This subsection is intended to give only an overview of the compression algorithm. Further details can be found in (Mantiuk, Efremov, Myszkowski & Seidel 2006a) or (Mantiuk, Efremov, Myszkowski & Seidel 2006b) and on the project web page: `http://www.mpii.mpg.de/resources/hdr/hdrmpeg/`.

We implemented and tested a dual video stream encoding for the purpose of a backward compatible HDR encoding, however, we believe that other applications that require encoding multiple streams can partly or fully benefit from the proposed method. For example, a movie could contain a separate video stream for color blind people. Such a stream could be efficiently encoded because of its high correlation with the original color stream. Movie producers commonly target different audiences with different color appearance (for example *Kill Bill 2* was screened with a different color stylization in Japan). The proposed algorithm could be easily extended so that several color stylized movies could be stored on a single DVD. This work is also a step towards an efficient encoding of multiple viewpoint video, required for 3D video (Matusik & Pfister 2004).

4 Conclusions

In this paper we introduce the concept of HDR imaging, pointing out its advantages over the traditional digital imaging. We describe our implementation of the image processing software that operates on HDR images and offers flexibility necessary for research purposes. We believe that the key issue that needs to be resolved to enable wide acceptance of HDRI is efficient image and video compression of HDR content. We address the compression issues by deriving a perceptually-motivated HDR color space capable of encoding the entire dynamic range and color gamut visible to the human eye. We propose also two compression algorithms, one being a straightforward extension of the existing MPEG standard, and the other offering backward compatibility with traditional video content and equipment. The proposed backward-compatible algorithm facilitates a smooth transition from the traditional to high-fidelity HDR DVD content.

In our work we try to realize the concept of an imaging framework that would not be restricted by any particular imaging technology and, if storage efficiency is required, be limited only by the capabilities of the human visual system. If the traditional imaging is strongly dependent on the particular technology (e.g. primaries of color spaces based on the red, green and blue phosphor in CRT displays), HDRI can offer an image-independent representation of images and video. However, redesigning existing imaging software and hardware to work with HDR content requires a lot of effort and definition of new imaging standards. Our mission is to popularize the concept of HDR imaging, develop standard tools and algorithms for processing HDR content and research the aspects of human perception that have key influence on digital imaging.

Acknowledgements

I would like to thank my advisors, Karol Myszkowski and Hans-Peter Seidel, for supporting my work on HDRI. Special thanks go to Grzegorz Krawczyk and Alexander Efremov for their work on the HDR video compression projects.

References

CIE (1981). *An Analytical Model for Describing the Influence of Lighting Parameters Upon Visual Performance*, Vol. 1. Technical Foundations, CIE 19/2.1, International Organization for Standardization.

DICOM PS 3-2004 (2004). Part 14: Grayscale standard display function, *Digital Imaging and Communications in Medicine (DICOM)*, National Electrical Manufacturers Association. **URL:** *http://medical.nema.org/dicom/2004.html*

Krawczyk, G., Myszkowski, K. & Seidel, H.-P. (2005). Perceptual effects in real-time tone mapping, *SCCG '05: Proc. of the 21st Spring Conference on Computer Graphics*, pp. 195–202.

Lubin, J. & Pica, A. (1991). A non-uniform quantizer matched to the human visual performance, *Society of Information Display Int. Symposium Technical Digest of Papers* (22): 619–622.

Mantiuk, R., Efremov, A., Myszkowski, K. & Seidel, H.-P. (2006a). Backward compatible high dynamic range mpeg video compression, *ACM Transactions on Graphics* **25**(3).

Mantiuk, R., Efremov, A., Myszkowski, K. & Seidel, H.-P. (2006b). Design and evaluation of backward compatible high dynamic range video compression, *MPI Technical Report MPI-I-2006-4-001*, Max Planck Institute für Informatik.

Mantiuk, R., Krawczyk, G., Myszkowski, K. & Seidel, H.-P. (2004). Perception-motivated high dynamic range video encoding, *ACM Transactions on Graphics* **23**(3): 730–738.

Mantiuk, R., Myszkowski, K. & Seidel, H.-P. (2006). Lossy compression of high dynamic range images and video, *Proc. of Human Vision and Electronic Imaging XI*, Vol. 6057 of *Proceedings of SPIE*, SPIE, San Jose, USA, p. 60570V.

Matusik, W. & Pfister, H. (2004). 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes, *ACM Trans. on Graph.* **23**(3): 814–824.

Poynton, C. (2003). *Digital Video and HDTV: Algorithms and Interfaces*, Morgan Kaufmann.

Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentacoste, M., Ghosh, A. & Vorozcovs, A. (2004). High dynamic range display systems, *ACM Trans. on Graph.* **23**(3): 757–765.

Sezan, M., Yip, K. & Daly, S. (1987). Uniform perceptual quantization: Applications to digital radiography, *IEEE Transactions on Systems, Man, and Cybernetics* **17**(4): 622–634.

Nominiert für den Heinz-Billing-Preis 2006

ISD – A Bayesian Software for NMR Structure Determination

Wolfgang Rieping
Michael Habeck

Abstract

Structure determination by NMR is often perceived as being less objective than x-ray crystallography. The major reason for this is the lack of an accepted measure of the quality of an NMR structure, and the use of empirical rules for deriving geometrical constraints from the experimental data. To alleviate this problem, we have developed ISD (Inferential Structure Determination), a probabilistic framework for structure determination. ISD uses Bayesian inference to derive a probability distribution that represents the unknown structure and its uncertainty. This probability distribution also determines additional unknowns, such as theory parameters, that previously had to be chosen empirically. Here we describe a new software package that implements this methodology. The program uses parallel Markov chain Monte Carlo sampling techniques to search for probable structures and parameter sets. Our software is unique in its capability to perform a fully probabilistic structure determination, and has proven superior to standard methods in providing an objective figure of merit and improving structural quality.

The program comes with a free academic license and is available via the ISD web site at http://www.bioc.cam.ac.uk/isd

1 Introduction

In an aqueous environment, proteins fold into thermodynamically stable three-dimensional structures. A detailed understanding of the biological function of proteins or DNA requires knowledge of its molecular structure. It is also

crucial in applications such as drug design. High-resolution nuclear magnetic resonance (NMR) spectroscopy is, besides X-ray crystallography, the routine method for determining biomolecular structures with atomic resolution [1]. In comparison to X–ray crystallography, solution NMR allows the study of proteins in their natural environment, and also provides a dynamical picture of the molecules. But NMR structure determination is far from being straightforward: It requires several manual or semi-automatic preprocessing steps such as spectral analysis, peak picking, and resonance assignment. In the final step, geometrical constraints are derived which are then used to calculate the molecular structure.

Each of these stages requires human intervention, which is why structure determination by NMR is often perceived as being less objective than X–ray crystallography. The major reasons for the "subjectiveness" of NMR structures are: (1) the lack of a generally accepted measure for assessing the quality of an NMR structure, (2) the use of heuristics and rules of thumb in the derivation of geometrical constraints. Strictly speaking, a protein structure determined from experimental data is only useful if it is accompanied by some measure of reliability. Recent works discussing errors in published NMR structures [2] highlight the danger of subjective elements in structure determination procedures.

The aforementioned problems have a common source: Structure determination requires reasoning from incomplete information which is why protein structures necessarily remain uncertain to some degree. Existing methods, however, are based on the concept of structural constraints, and are therefore incapable of taking this uncertainty into account. In essence, ISD relies on Bayesian probabilistic inference that represents any uncertainty through probabilities which are then combined according to the rules of probability calculus. The application of this approach is computationally demanding, and has become feasible only recently due to the development of efficient stochastic sampling algorithms (Markov chain Monte Carlo methods) and increased computational power provided by computer clusters.

2 Inferential structure determination

The principal difficulty in structure determination by NMR is the lack of information required to unambiguously reconstruct a protein structure. Conventional methods view structure determination as a minimisation problem: A so-called "hybrid energy" function combines a pseudo energy term that incorporates the experimental constraints with a force field describing the physical interactions between the atoms. Minimising the hybrid energy is

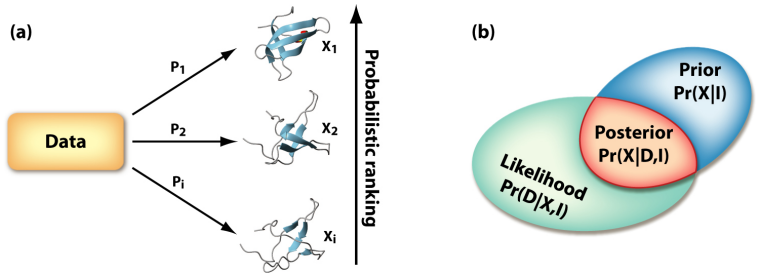


Fig. 1: Probabilistic ranking and Bayes' theorem. The experimental data are used to rank every conformation of a protein in terms of a probability (a), i.e. we do not derive geometrical constraints that would completely rule out structures. If of two conformations one has higher probability, then it is more supported by the data. The spread of the probability distribution reflects how well we can determine a structure from the available information. If only a single conformation has non-zero probability, the data uniquely determine the structure. If the probabilities are constant, the available data is uninformative with respect to the structure. Realistic cases lie somewhere in between. Bayes' theorem (b) combines prior information with experimental evidence, represented in terms of a likelihood function, in a consistent way. The posterior distribution represents everything that can be said about the molecular structure given the data and our prior knowledge.

then assumed to answer what the "true" structure of a molecule is. This rule, however, implicitly assumes that there is a unique answer. Repeating the minimisation procedure multiple times, as is standard practice in conventional approaches, does not adequately represent the ambiguity and makes it difficult to judge the validity and precision of NMR structures in an objective way.

We have argued that it is a misconception to use structure calculation methods that are only appropriate if the objective is to obtain a unique structure. Instead, we view structure determination as an *inference problem* [3, 4], requiring reasoning from incomplete and uncertain information. In contrast to conventional methods, we do not convert the data into geometrical constraints, but use them directly to rank all possible conformations of the molecule. Quantitatively, such a ranking requires us to assign a probability P_i to every protein conformation X_i [5] (Fig. 1a). We demand the probabilities to be objective in the sense that they should depend only on the data and on relevant prior information (such as the theoretical models to describe the data or knowledge about physical interactions). Thus we are dealing with a conditional probability, $\Pr(X|D, I)$, quantifying how likely a certain conformation X is the correct structure given the data "D" and background information "I". Any inferential structure determination is solved by exploring this probability distribution.

But how can we set up $\Pr(X|D, I)$ for a concrete structure determination

problem? The answer comes from Bayes' theorem [6] which states that the solution to any structure determination problem is proportional to the product of the likelihood of the data given a structure, $\Pr(D|X,I)$, and the prior probability $\Pr(X|I)$ (Fig. 1b). That is, once we are able to write down the likelihood and the prior distribution for a particular structure determination problem, we simply use Bayes' theorem to obtain a relative probability for every conformation of the protein, and thus solve the ranking problem. At first glance this seems to complicate things even further, but it turns out that $\Pr(D|X,I)$ and $\Pr(X|I)$ are relatively easy to set up.

AN EXAMPLE Let us consider a concrete example. The most informative class of NMR observations are based on the Nuclear Overhauser Effect (NOE) [7]. The NOE is a relaxation effect that leads to an NMR signal with an intensity I roughly proportional to the inverse sixth power of the distance d between two nuclear spins. However, the model $I \propto d^{-6}$ neglects dynamics [8] and spin diffusion effects [9]. Due to these theoretical limitations and experimental noise, observed NOE intensities can not be predicted with certainty from a protein structure. In order to deal with deviations between observations and predictions, we introduce an error parameter σ that quantifies how closely our predictions match the observations. Since intensities and distances are positive quantities, we model their deviations with a log-normal distribution [10]. If we have a whole set of intensities I_i with corresponding distances d_i the likelihood of the data is a product of log-normal distributions:

$$\Pr(D|X,\alpha,\sigma,I) = \prod_i \frac{1}{\sqrt{2\pi}\sigma I_i} \exp\left\{-\frac{1}{2\sigma^2}\left[\log I_i - \log(\alpha d_i^{-6})\right]^2\right\} \quad (1)$$

where α is the unknown proportionality factor, and the distances depend on the protein conformation, i.e. $d_i = d_i(X)$.

This example illustrates two points: First, it is straightforward to write down the likelihood function. Second, one typically needs to introduce auxiliary parameters, such as σ and α, that are necessary to describe the measurements, but cannot be determined experimentally. In standard methods, such parameters need to be set empirically, which can bias the results, and adds to the problem of structure validation [11, 12]. In Bayesian theory, such *nuisance parameters* are treated in the same way as the coordinates: They are estimated from the data by applying Bayes' theorem on the joint parameter space.

Bayes' theorem requires the assignment of prior probabilities for the conformational degrees of freedom and the nuisance parameters. Using arguments from statistical physics, it turns out that $\Pr(X|I)$ is the canonical ensemble:

$$\Pr(X|I) \propto \exp\{-\beta E(X)\} \quad (2)$$

where $E(X)$ is a molecular force field encoding chemical information on bond lengths, bond angles, etc.; β is the inverse temperature. In the simplest case, the prior probabilities for the nuisance parameters are uniform distributions, and the posterior distribution for *all* unknowns is:

$$\Pr(X, \alpha, \sigma | D, I) \propto \alpha^{-1} \sigma^{-n-1} \exp\Big\{ -\beta E(X) - \frac{1}{2\sigma^2} \sum_i \log^2 (I_i / \alpha d_i^{-6}) \Big\}. \tag{3}$$

Hence, probability calculus formally solves the problem of structure determination from NOE data: The posterior probability distribution represents the complete information on the possible conformations of the molecule, as well as on the values of our nuisance parameters, α and σ in this case.

3 Algorithm

For realistic problems, the posterior probability is a very complex mathematical function. It is defined over a space of typically several hundred dimensions, which makes it impossible to visualise the posterior probability directly or to analyse it analytically. Therefore, we need to employ numerical methods. A conceptually simple way to investigate a high dimensional probability is to draw samples from it in such a way that the distribution of samples follows this probability. The samples can then be used to estimate most likely parameters, averages, variances, etc. [13]. Therefore in the ISD framework, structure calculation amounts to the generation of random samples from the joint posterior probability $\Pr(X, \alpha, \sigma | D, I)$ where α and σ denote the nuisance parameters and errors, respectively. This differs fundamentally from conventional structure calculation algorithms because the uncertainty of the structure is explicitly taken into account, and nuisance parameters are not kept fixed.

3.1 Gibbs sampling

We generate posterior samples $(X^{(k)}, \alpha^{(k)}, \sigma^{(k)})$ by using a hierarchical "Gibbs sampling" scheme that combines several Markov chain Monte Carlo strategies. The Gibbs sampling procedure [16] facilitates a split-up of the sampling problem into several steps. Each parameter class, X, α, and σ, is sampled sequentially conditioned on the current values of the other parameters (Fig. 2). In order to apply a Gibbs sampling scheme, we need to be able to simulate the conditional posterior densities for the nuisance parameters and the coordinates. For simple distributions, e.g. a normal or log-normal distribution, this can be done by using random number generators. To sample the highly correlated conformational degrees of freedom, however, we need

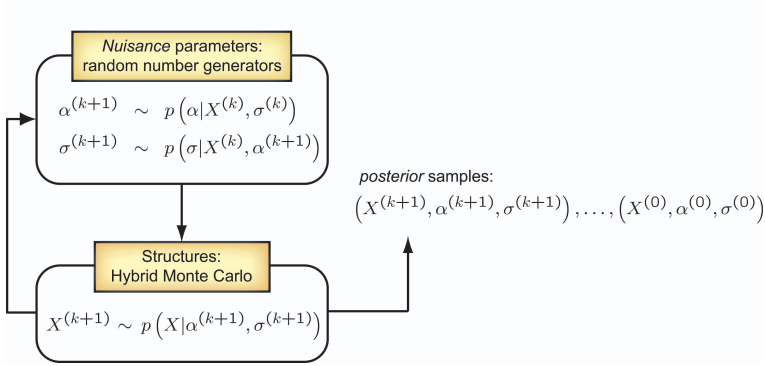


Fig. 2: Gibbs sampling scheme used to generate samples from the posterior probability for protein conformation X and nuisance parameters α and σ. Gibbs sampling is an iterative scheme that, upon convergence, produces samples from the full posterior distribution. The nuisance parameters can directly be drawn from their posterior probabilities. To update the conformational degrees of freedom we employ the HMC algorithm. This algorithm uses molecular dynamics [14] to generate a candidate conformation which is accepted according to the Metropolis criterion [15]. The molecular dynamics is defined by the negative log-posterior probability with fixed nuisance parameters.

to employ more elaborate techniques such as the Hybrid Monte Carlo (HMC) method [17].

3.2 Replica-exchange Monte Carlo

For complex systems such as proteins, the Gibbs sampler is likely to get trapped in one of the modes of the posterior distribution and thus fails to explore the entire parameter space. These modes correspond to protein conformations that all fulfill the data well. Missing a high-probability fold would bias our analysis.

A physical system trapped in a metastable state can be melted by increasing its temperature. For sufficiently high temperatures the system easily explores all regions of the configuration space. The Replica-exchange Monte Carlo method [18] exploits this observation: it considers a composite Markov chain comprising several non-interacting copies of the system, so-called "replicas", each of which is simulated at a different temperature. Neighbouring replicas are coupled by exchanging configurations after a number of Gibbs sampling steps ("super-transition"), which significantly enhances the mobility of the individual Markov chains. We have extended this scheme by introducing two generalized "temperatures", λ and q, to independently control the likelihood function and the prior probability [19] (Fig. 3). The parameter λ weighs the likelihood function, and thus controls the influence of the data. We further

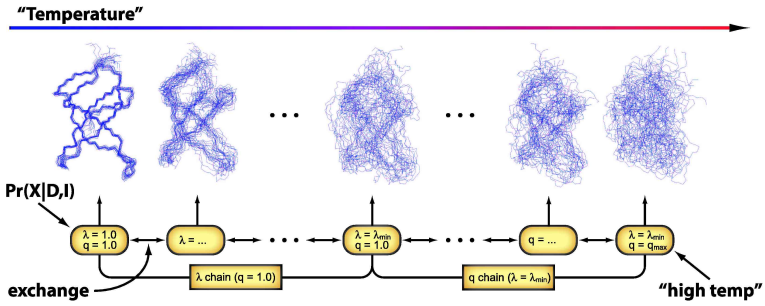


Fig. 3: Replica-exchange Monte Carlo algorithm. We embed the Gibbs sampler (Fig. 2) in a Replica-exchange Monte Carlo scheme which simulates a sequence of "heated" replicas of the system. Two generalized temperatures, λ and q, control the shape of the likelihood function and of the prior distribution, respectively. For $\lambda = 1$ the data are switched on, for $\lambda = 0$ they are switched off. For $q = 1$, the canonical ensemble is restored as prior probability [cf. Eq. (2)]. For $q > 1$ physical interactions are gradually switched off and the prior probability approaches a flat distribution over conformation space. We arange the replicas in such a way that first the data are switched off (by gradually decreasing λ). In the other half of the arrangement, we additionally switch off the physical interactions by increasing q.

improve the sampling by replacing the canonical ensemble with Tsallis generalized ensemble [20, 21, 22, 23]. Tsallis ensemble is based on a non-linear transformation of the force field $E(X)$ and involves a parameter q which controls the strength of the non-linearity. The transformation is chosen such that high energy configurations are no longer exponentially suppressed, but follow a power law. This has the effect that atoms can exceedingly pass through each other, thus facilitating large conformational changes. During a simulation, states diffuse up and down in the replica arrangement, which guarantees ergodic sampling of the posterior distribution.

4 The ISD software package

The computer program ISD implements the methodology outlined in the preceeding sections in the form of an object-oriented software library. ISD is written in the programming languages Python [24] and C. Python is one of the most advanced object-oriented programming languages. It is open source, offers strong support for integration with other languages, and is easy to learn. C is well known for its excellent performance.

Time critical routines, e.g. for computing the energy of physical interactions within a molecule, are coded in C for optimal performance. So-called "wrappers" glue the C to the Python world thus enabling us to use C functions seamlessly from within Python. Using this hybrid approach, we benefit from

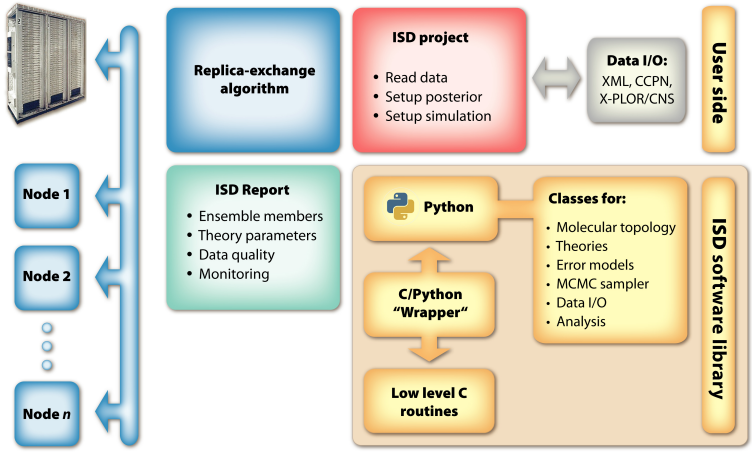


Fig. 4: Overview of the architecture of ISD. On the user-side, ISD (red) manages the import of the experimental data (grey), the setup of a replica-exchange calculation on a computer grid (blue), as well as the generation of a report containing analyses of the calculation results (green). The ISD software library (amber) provides the functional basis for performing these steps.

both Python's advanced language design, and the performance of C.

WEB SITE AND DISCUSSION GROUP The program comes with a free academic license and is available via the ISD web site at

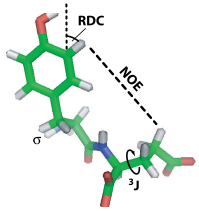
http://www.bioc.cam.ac.uk/isd

The web site also provides related topics and supplemental information on ISD, such as a manual. Users that are interested in the program or that would like to share their experience with other users, can join our user group at

http://groups.google.com/group/isd-discuss

4.1 Overview

Figure 4 shows the principal design of ISD. The object-oriented software library shown in amber forms the heart of ISD. It provides the functionality needed for setting up a project, performing the structure calculation, as well as analysing the results. Each of these steps is conducted by a "project file" depicted in red. The project comes with default parameter settings which have proven suitable for inferring biomolecular structures based on various NMR experiments. The experimental data are incorporated into a calculation by setting-up a likelihood function for every data set. The program does all these steps fully automatically – the user only needs to specify information on the type and location of his data.



Parameter	Structure	Theory	Error	n†
NOE intensity	distance	ISPA	Log-normal	2
Scalar coupling	torsion angle	Karplus curve	Normal	4
RDC	bond orientation	Saupe tensor[1]	Normal	6
Chemical shift	torsion angle	Talos[2]	Von-Mises[3]	1
Distance	distance	None	Log-normal	2
Hyrdogen bond	distance	None	Log-normal	1
Disulfide bridge	distance	None	Log-normal	0
Dihedral angle	torsion angle	None	Von-Mises	1

Tab. 1: Experimental NMR parameters. Left panel: Relationship between NMR parameter and structural parameter. Right panel: NMR parameters supported by ISD. †Number of nuisance parameters to be estimated for a particular data set. [1]RDCs depend on the orientation of the associated bond angle with respect to an external frame of reference. Experiments establish this reference by using specific media to partially align the molecules. The Saupe tensor describes the alignment. [2]Talos[25] is computer program that predicts backbone torsion angles from chemical shifts σ. [3]The von-Mises distribution is the "normal distribution for periodic variables".

4.2 Supported NMR parameters

The program supports most of the commonly used experimental NMR parameters (cf. Tab. 1). The NOE contains information about the spatial vicinity of protons and is the most important source of structural information. It has been utilised to determine virtually all of the NMR structures currently stored in the Protein Data Bank (PDB) [26]. Residual dipolar couplings (RDC) are less informative than the NOE but can be very useful for obtaining information on the global fold of the protein. Three-bond scalar couplings are routinely measured to obtain precise information on the local conformation of a biomolecule [27]. In addition to experimental parameters, the user can specify structural information directly in the form of distances and torsion angles.

4.3 Theories

To incorporate experimental data into a calculation, a "theory" is used to calculate the ideal value of a measurand. The ideal value depends on the three-dimensional coordinates of a structure and, depending on the theory, a set of theory parameters. The program also supports geometric parameters directly, such as distances or dihedral angles.

4.3.1 NOE intensities

ISD uses the isolated spin pair approximation to calculate experimental NOE intensities from the three-dimensional coordinates of a structure. In case the NOE involves atoms that are part of equivalent groups (basically methylene

and isopropyl groups), the partial NOE volumes are added up, and the observed NOE intensity $I_{\exp}$ is calculated as

$$I_{\exp}(X) = \gamma \sum_{i<j} d_{ij}(X)^{-6},$$

where X denotes the three-dimensional coordinates of the structure, $d_{ij}(\cdot)$ the distance between atoms i and j, and γ the scale of the measured intensities. The scale γ is a typical theory parameter: It cannot be determined experimentally but is required in order to match calculated and measured values. During the course of a calculation, the scale is estimated from the data.

4.3.2 Scalar couplings

The Karplus curve is used to describe the observed three-bond scalar coupling constants ${}^3J$ in terms of the intervening torsion angle φ:

$${}^3J(\varphi) = A\cos^2\varphi + B\cos\varphi + C.$$

The coefficients A, B, C of the Karplus curve cannot be determined experimentally and, therefore, need to be estimated from the data (see [27] for details).

4.3.3 Residual dipolar couplings

The Saupe or alignment tensor $\mathbf{S}$ is used to describe the observed dipolar couplings D in terms of the inter-atomic bond vector $\mathbf{r}$:

$$D = \mathbf{r}^T \mathbf{S} \mathbf{r}. \tag{4}$$

The alignment tensor is symmetric and trace-less, $\mathbf{S}^T = \mathbf{S}$, $\mathrm{tr}[\mathbf{S}] = 0$, and can be parametrized using five independent elements $s_1, \ldots, s_5$. The explicit parameterization of the alignment tensor is:

$$\mathbf{S} = \begin{pmatrix} s_1 - s_2 & s_3 & s_4 \\ s_3 & -s_1 - s_2 & s_5 \\ s_4 & s_5 & 2s_2 \end{pmatrix}.$$

The alignment tensor describes the average orientation of the molecule in the alignment medium and also quantifies the degree of alignment. The average orientation can be calculated by diagonalizing the tensor: $\mathbf{S} = \mathbf{U}\mathbf{L}\mathbf{U}^T$. The rotation matrix $\mathbf{U}$ describes the average orientation. The eigenvalues of $\mathbf{S}$ (i.e. the elements of the diagonal matrix $\mathbf{L}$) can be transformed into an axial and rhombic component of the alignment tensor.

The tensor elements cannot be determined experimentally and, therefore, need to be estimated from the data. Several heuristics such as the histogram method [28] have been developed that allow for an approximate estimation of the axial and rhombic components from the dipolar coupling data alone. When using ISD, such preliminary analyses are superfluous, because the unknown tensor elements are treated as nuisance parameters and estimated during the actual structure calculation.

4.4 Error models

Measured and calculated NMR parameters never match. Deviations of measured from calculated data are the result of experimental noise and, often more important, approximations in the theory used to calculate the data from the three-dimensional coordinates of a structure. For example, most expressions for calculating NMR parameters neglect the dynamics of a molecule, which can lead to systematic deviations of calculated from measured values. The magnitude of these deviations is a priori unknown. In a probabilistic framework, this lack of knowledge is described by an error model. ISD supports various error models, which shall be described in the following paragraphs.

4.4.1 Log-normal

The lognormal distribution can be considered as the "Gaussian for positive quantities" and is better suited for describing positive measurands (such as distances or NOE intensities) than, for example, a Gaussian or a probability distribution that corresponds to a flat-bottom potential (see [10] for details). The density function of a lognormal distribution is

$$\Pr(A_{\mathrm{exp}}|A_{\mathrm{calc}}(X),\sigma) = \frac{1}{\sqrt{2\pi}\sigma A_{\mathrm{exp}}} \exp\left\{-\frac{1}{2\sigma^2}\log^2\frac{A_{\mathrm{exp}}}{A_{\mathrm{calc}}(X)}\right\},$$

where A_{exp} and $A_{\mathrm{calc}}(X)$ denote the observed and calculated NMR parameter, respectively. The error parameter σ relates to the width of the distribution. As mentioned above, σ quantifies the degree to which the data can be explained on the basis of a single structure, and a given theory. However, σ is a priori unknown and needs to be estimated from the data.

4.4.2 Gaussian

ISD uses a normal distribution to model the errors of scalar and dipolar coupling constants. The magnitude of these errors is a priori unknown, and is described by an error parameter σ which relates to the width of the distribu-

tion. The density function of the normal distribution is

$$\Pr(A_{\mathrm{exp}}|A_{\mathrm{calc}}(X),\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(A_{\mathrm{exp}} - A_{\mathrm{calc}}(X))^2\right\},$$

where A_{exp} and $A_{\mathrm{calc}}(X)$ denote the observed and calculated NMR parameter, respectively.

4.4.3 Von Mises

The von Mises distribution is the equivalent of the normal distribution for periodic variables. ISD uses this distribution to model the deviations of measured from calculated torsion angles. These deviations are the result of experimental noise and errors in predicted torsion angles (e.g. if they are predicted from chemical shifts). The magnitude of the deviations is a priori unknown, and is described by a shape parameter κ which quantifies the precision of the torsion angle restraints. The density function of the von Mises distribution is

$$\Pr(\varphi_{\mathrm{exp}}|\varphi_{\mathrm{calc}}(X),\kappa) = \frac{1}{2\pi I_0(\kappa)}\exp\left\{\kappa\cos(\varphi_{\mathrm{exp}} - \varphi_{\mathrm{calc}}(X))\right\},$$

where φ_{exp} and $\varphi_{\mathrm{calc}}(X)$ denote the observed and calculated torsion angle, respectively. I_0 is the Bessel function of the first kind. If the estimated shape parameter κ has a negative sign, this indicates that a common phase of π needs to be added to the torsion angles to obtain the best fit.

4.5 Supported data formats

ISD represents experimental data via a data format based on the eXtensible Markup Language (XML) [29]. XML allows definition of portable and human readable formats for information exchange and supports document validation, thus guaranteeing consistency and integrity of the data. The program provides XML data formats for describing the molecular topology (ISD uses the IUPAC nomenclature standard [30]) and the experimental NMR parameters described above. Additional to XML, the data can also be specified in X–PLOR/CNS .tbl format (for distances and dihedral angles), and TALOS format (dihedral angles). During the initialisation of a calculation proprietry data formats are automatically converted into ISD XML format.

4.6 Replica-exchange calculation on a computer cluster

Setting up a replica-exchange simulation for calculating a molecular structure only requires specifying the generalised temperature values (λ, q), as well as

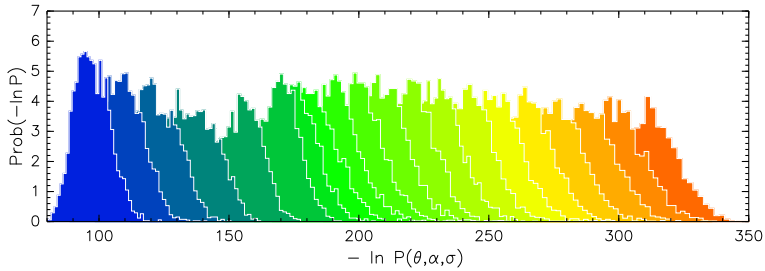


Fig. 5: Distributions of the negative log-posterior probabilities of the replicas. The color encodes the position in the replica chain (i.e. "temperature"). Blue: true posterior distribution, red: "high-temperature" posterior distribution.

the number of Gibbs sampling steps per super-transition. The default settings are suited for standard systems, however depending on the system size and the quality of the data, "fine-tuning" the temperatures can improve the convergence of the simulation.

As described in section 3.2, each super-transition requires a number of Gibbs sampling updates for every replica. Because communication between the replicas is rare (it is only required upon the exchange of states), a replica-exchange scheme is particularily suited to be run on multiple machines in parallel. The setup of a parallel simulation is straightforward: All the user needs to do is specify a list of available machines. ISD then launches the required services on each of these machines. During a calculation every slave process receives requests from the master to perform a number of Gibbs sampling steps at a given temperature. ISD then collects the results, and starts a new super-transition. The parallel setup is completely transparent to the user, since the program runs locally on his desktop machine. The mixing efficiency of the replica algorithm depends on the number of, and the rate of exchange between the replicas. Therefore, the temperatures λ and q need to be chosen carefully as they control the overlap between the posterior distributions and hence the rate of exchange. The default settings of our program typically result in an average rate of exchange of 70% (Fig. 5). For optimal performance, the number of available machines should be larger than the number of replicas, which defaults to 50.

4.7 Analysis of the calculation results

ISD stores all calculation results in Python's persitent object format ("Python pickles"). Users familiar with Python can thus access the full information generated during a calculation to perform further analyses. How this is done is discussed in detail in the manual, which is available on the ISD web site.

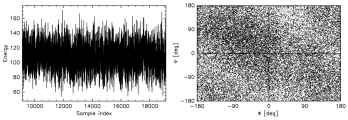


Figure 3: Left: trace of the total energy associated with the target distribution. For converged calculations the energy should scatter around its median (dotted). Right:? ? plot for all residues of the high temperature ensemble. For an appropriate? q schedule this distrubtion should be uniform.

to analyse a particular NMR experiment. It also depends, to a lesser extent, on approximations in the background assumptions, such as the form of the force field used to describe physico-chemical interactions between the atoms of a molecule. In others, incomplete data always leads to uncertain structures, as common sense would suggest.

2.1 Uncertainty

It is important to keep in mind that structural uncertainty should always be regarded in the sense of an "error bar" of a structure. This error bar is of statistical nature, that is there is no causal connection to real physical fluctuations of the atom posititons. Physical fluctuations can be one of the reasons why a structure is uncertain ?, uncertainty and fluctuations might even correlate on a qualitative level. Quantitatively, however, we cannot infer dynamics from structural uncertainty.

The uncertainty of the present structure, quantified via the median uncertainty in the position of its $C_\uparrow$ atoms (s. Figure 4) has been calculated on the basis of 100 ensemble members and amounts to $?_{struct} = 1.44 \pm 0.66$ Å. The directory /WORKINGPATH/analysis/structures contains 100 members of the structure ensemble (stored in IUPAC PDB format).

2.2 Quality scores

In order to validate the structure further, we calculate a number of quality scores using the programs WhatIf and Procheck. The scores are based on our knowledge about general geometric features of proteins, and are derived by comparing the three-dimensional coordinates of the calculated structures with a database of high-resolution x-ray structures. The scores (see table 2) include measures of the packing of a protein, of the local geometry, as well as of the compatibility of the backbone with known protein structures.

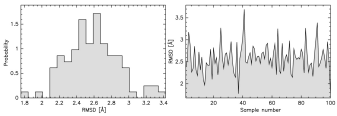


Figure 6: Distribution and trace of the RMSD to the reference structure, calculated for the atoms CA, C, N, O. The most probable structure has an RMSD of 1.78 Å.

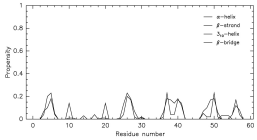


Figure 7: Secondary structure assignment. Shown is the propensity with which a residue adopts the secondary structural states ?-helical, ?-strand, 3_{10}-helical or isolated?-bridge. Secondary structure assignments were derived with the program DSSP, propensities are calculated on the basis of the conformational ensemble.

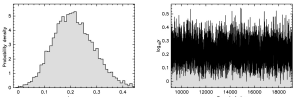


Figure 8: Probability distribution (left) and trace (right) of the scale ("calibration factor") ? used in the ISPA to relate measured peak intensities to distances.

Fig. 6: Excerpt of an ISD report. An ISD report summarises the calculation results in the form of a PDF document. It provides graphical information on the performance of a calculation, as well as various analyses, such as of the data quality and estimates of all theory parameters.

A more convenient way of accessing the simulation results is to create an ISD report. An ISD report summarises the calculation results and is formatted in the portable document format (PDF) (see Figure 6). The report provides graphical information on the performance of the calculation, analyses of each data set (for example, an estimate of its quality). It also gives estimates of all theory parameters, and creates PDB files containing the coordinates of representative members of the probabilistic structure ensemble. Furthermore, it provides results of validation checks performed with the computer programs WhatIf [31] and Procheck [32].

5 Applications

The ISD software package has proven powerful to solve a number of difficult structure determinations from NMR data. In addition to reconstructing a protein structure with a measure of local uncertainty, the program is able to estimate parameters that typically need careful manual optimization and thus allows for an objective interpretation of the data. In the following, we will discuss some of these aspects for two examples: a sparse data set measured for the SH3 domain of the protein Fyn [35] and a complete data set comprising ^{15}N and ^{13}C measurements for the Tudor domain [36].

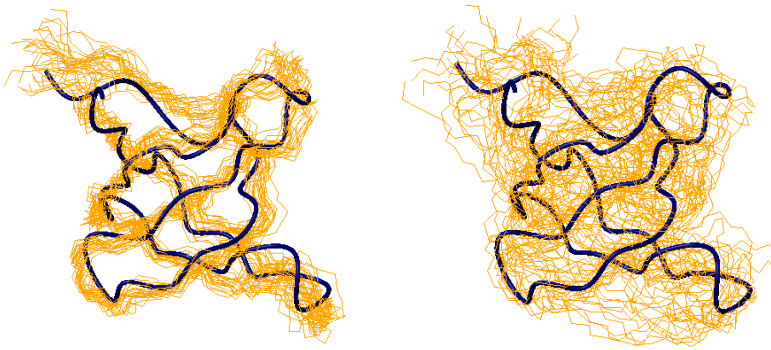


Fig. 7: Structure ensembles. Calculated structures (orange) were superimposed onto the crystal structure of the SH3 domain (blue). Left panel: 20 most likely conformations obtained with ISD. Right panel: For comparison, a conventional structure ensemble was calculated by repeatedly running a standard minimisation protocol [33, 34]. The root mean square deviation (rmsd) between the backbone atom coordinates of the crystal structure and the ISD structure is 1.84 $\pm$ 0.20 Å across the entire protein and 1.36 $\pm$ 0.19 Å for the secondary structure elements. This is a significant improvement over standard techniques: The conventional structure ensemble has an rmsd of 3.07 $\pm$ 0.53 Å for all residues and 1.93 $\pm$ 0.34 Å for secondary structure elements.

5.1 Structure determination from a sparse data set

A surprising result is that a probabilistic approach is able to extract more information from sparse data. This was demonstrated [3] for the SH3 data that comprise 154 experimental distances derived from NMR experiments on a deuterated protein sample. Typically, ten times larger data sets are required to determine a protein structure. We used the standard setup described in the previous section to sample protein structures from the posterior probability resulting from the data. The most pronounced features of the posterior probability can be represented by a bundle of structures (Fig. 7). Compared to conventional structure ensembles, the ISD ensemble is significantly better defined and systematically closer to the structure obtained with X-ray crystallography [37].

5.2 Objective figures of merit

Because conventional structure ensembles depend on user-specific parameter settings and on the minimization protocol, it is difficult if not impossible to assign statistically meaningful error bars to atomic coordinates. In contrast, stochastic samples drawn from the full posterior probability $\Pr(X, \alpha, \sigma|D, I)$

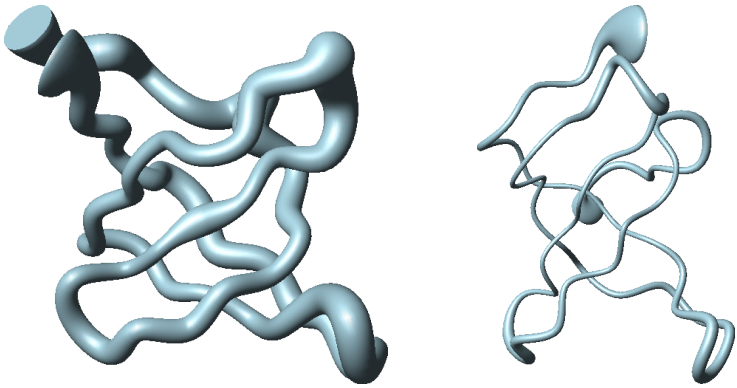


Fig. 8: Conformational uncertainty. Left panel: The 20 most probable conformations (also shown in the previous figure) from the sparse SH3 data set were used to calculate the average structure and its local precision. The local precision ranges from 0.6 Å for secondary structure elements to 4.6 Å for loop regions (bottom and right hand side) and termini (top). The average precision is 1.07 Å. Right panel: Average structure and local precision for an analysis of the Tudor domain [4]. In this calculation, the number of data is approximately ten times larger, which leads to a much better defined structure. This result accords with common sense.

are statistically well defined and can directly be used to calculate estimates of mean values and standard deviations. In consequence, we can derive an average structure with atom-wise error bars and are thus able to define an objective figure of merit for NMR structures (Fig. 8).

An important quantity are the experimental errors σ required by the probabilistic model for the data. Figure 9 demonstrates that the error of the data can be estimated along with a structural model. It is no longer necessary to determine the error empirically [38]. Hence, our conformational samples are not biased by additional heuristics that can depend on the person analysing the data, but are objective in the sense, that they exclusively reflect the information content of the data and the assumptions made in the probabilistic models.

Conclusion

Bayesian probability theory is well-suited to formalise and solve macromolecular structure determination problems. Our software package ISD successfully demonstrates the feasibility of a fully Bayesian analysis of NMR data by means of Markov chain Monte Carlo sampling. ISD is unique in its capa-

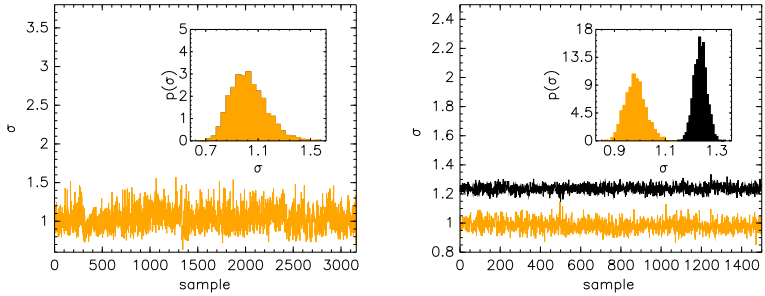


Fig. 9: Posterior samples of the errors in two structure calculations. Left panel: error of the SH3 data set. Right panel: errors of the Tudor data sets (black: ^{13}C data, orange: ^{15}N data). The insets show the corresponding histograms. The posterior samples vary around the most probable value. When compiling histograms from these samples, we obtain unimodal distributions. The error of the SH3 data set is less well defined than for the Tudor data, which is due to the much smaller number of data. Nevertheless, the SH3 and Tudor ^{15}N errors have approximately the same mean value, indicating that the data sets are of similar quality. The error of the Tudor ^{13}C data is significantly larger, which is due to relaxation mechanisms that affect the ^{13}C data more than the ^{15}N measurements.

bility to not only calculate the most likely conformation of a biomolecule, but also its "error bar". This is highly relevant in practical applications such as drug design, where it is essential to correctly interpret the structural details of a compound, which can be difficult if an objective figure of merit is missing.

Our program derives the molecular structure and its uncertainty on the basis of a mathematically closed expression, and therefore strictly separates algorithmic issues from data modelling. In contrast, the precision of the atomic coordinates calculated by conventional methods is subjective, since it largely depends on the properties of the minimization algorithm used to generate the structure, and on choices in data treatment prior to structure calculation.

A major advantage of a Bayesian approach is its ability to cope with nuisance parameters. Standard methods do not provide unique rules for obtaining optimal values, in particular not in the case of multiple nuisance parameters. Using our program, auxiliary quantities need not be chosen empirically but are automatically estimated along with the atomic coordinates. Hence, tedious and time-consuming searches for suitable values are no longer necessary. Once the theory to describe the data has been chosen, probability calculus uniquely determines optimal values for all unknowns, and furthermore provides us with the uncertainty of the estimates.

It is straightforward to use inferential structure determination for combining NMR data with other kinds of structural information. This could include, for example, diffraction data from X-ray crystallography or information derived from evolutionary related proteins. On the technical side, the user ben-

efits from the open and flexible architecture of our software library. This is achieved by using the object-oriented programming language Python which facilitates a rapid incorporation of new experimental parameters into the existing design.

References

[1] K. Wüthrich. Nmr studies of structure and function of biological macromolecules (Nobel lecture). *Angew. Chem. Int. Ed. Engl.*, 42:3340–3363, 2003.

[2] S. B. Nabuurs, C. A. Spronk, G. W. Vuister, and G. Vriend. Traditional biomolecular structure determination by NMR spectroscopy allows for major errors. *PLoS Comput. Biol.*, 2:e9, 2006.

[3] W. Rieping, M. Habeck, and M. Nilges. Inferential Structure Determination. *Science*, 309:303–306, 2005.

[4] M. Habeck, M. Nilges, and W. Rieping. Bayesian inference applied to macromolecular structure determination. *Phys. Rev. E*, 72:031912, 2005.

[5] R. T. Cox. *The Algebra of Probable Inference.* John Hopkins University Press, 1961.

[6] E. T. Jaynes. *Probability Theory: The Logic of Science.* Cambridge University Press, Cambridge UK, 2003.

[7] I. Solomon. Relaxation processes in a system of two spins. *Phys. Rev.*, 99(2):559–565, July 1955.

[8] G. Lipari and A. Szabo. Model-free approach to the interpretation of nuclear magnetic resonance relaxation in macromolecules. 1. Theory and range of validity. *J. Am. Chem. Soc.*, 104:4546–4558, 1982.

[9] S. Macura and R. R. Ernst. Elucidation of cross relaxation in liquids by two-dimensional NMR spectroscopy. *Molecular Physics*, 41:95–117, 1980.

[10] W. Rieping, M. Habeck, and M. Nilges. Modeling errors in NOE data with a lognormal distribution improves the quality of NMR structures. *J. Am. Chem. Soc.*, 27:16026–16027, 2005.

[11] A. T. Brünger. The free R value: a novel statistical quantity for assessing the accuracy of crystal structures. *Nature*, 355:472–474, 1992.

[12] A. T. Brünger, G. M. Clore, A. M. Gronenborn, R. Saffrich, and M. Nilges. Assessing the quality of solution nuclear magnetic resonance structures by complete cross-validation. *Science*, 261:328–331, 1993.

[13] M. H. Chen, Q. M. Shao, and J. G. Ibrahim. *Monte Carlo Methods in Bayesian Computation*. Springer Verlag, Inc., New York, 2002.

[14] B.J. Alder and T.E. Wainwright. Studies in molecular dynamics. I. General method. *J. Chem. Phys.*, 31:459–466, 1959.

[15] N. Metropolis, M. Rosenbluth, A. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing. *J. Chem. Phys.*, 21:1087–1092, 1957.

[16] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. PAMI*, 6(6):721–741, 1984.

[17] S. Duane, A. D. Kennedy, B. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222, 1987.

[18] R. H. Swendsen and J.-S. Wang. Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Lett.*, 57:2607–2609, 1986.

[19] M. Habeck, M. Nilges, and W. Rieping. Replica-Exchange Monte Carlo Scheme for Bayesian Data Analysis. *Phys. Rev. Lett.*, 94:0181051–0181054, 2005.

[20] C. Tsallis. Possible Generalization of Boltzmann-Gibbs Statistics. *J. Stat. Phys.*, 52:479–487, 1988.

[21] U. H. E. Hansmann and Y. Okamoto. New Generalized-Ensemble Monte Carlo Method for Systems with Rough Energy Landscape. *Phys. Rev. E*, 56:2228–2233, 1997.

[22] U. H. E. Hansmann. Parallel tempering algorithm for conformational studies of biological molecules. *Chem. Phys. Lett.*, 281:140–150, 1997.

[23] T. W. Whitfield, L. Bu, and J. E. Straub. Generalized parallel sampling. *Physica A*, 305:157–171, 2002.

[24] G. van Rossum and J. de Boer. Linking a stub generator (AIL) to a prototyping language (Python). In *Proceedings of the Spring 1991 EurOpen Conference, Tromsø, Norway, May 20–24, 1991*, pages 229–247. EurOpen, 1991.

[25] G. Cornilescu, F. Delaglio, and A. Bax. Protein backbone angle restraints from searching a database for chemical shift and sequence homology. *J. Biomol. NMR*, 13(3):289–302, Mar 1999.

[26] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: a Computer-based Archival File for Macromolecular Structures. *J. Mol. Biol.*, 112:535–542, 1977.

[27] M. Habeck, W. Rieping, and M. Nilges. Bayesian estimation of Karplus parameters and torsion angles from three-bond scalar coupling constants. *J. Magn. Reson.*, 177:160–165, 2005.

[28] G. M. Clore, A. Bax, and A. M. Gronenborn. A robust method for determining the magnitude of the fully asymmetric alignment tensor of oriented macromolecules in the absence of structural information. *J. Magn. Reson.*, 133:216–221, 1998.

[29] The World Wide Web Consortium. Extensible Markup Language (XML) 1.0, W3C recommendation. `http://www.w3.org/TR/REC-xml`, 1999.

[30] J. L. Markley, A. Bax, Y. Arata, C. W. Hilbers, R. Kaptein, B. D. Sykes, P. E. Wright, and K. Wüthrich. Recommendations for the presentation of NMR structures of proteins and nucleic acids. *J. Mol. Biol.*, 280(5):933–952, 1998.

[31] G. Vriend. WHAT IF: a molecular modeling and drug design program. *J. Mol. Graph.*, 8:52–56, 1990.

[32] R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.*, 26:283–291, 1993.

[33] M. Nilges, M. J. Macias, S. I. O'Donoghue, and H. Oschkinat. Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from β-spectrin. *J. Mol. Biol.*, 269:408–422, 1997.

[34] A. T. Brünger, P. D. Adams, G. M. Clore, W. L. DeLano, P. Gros, R. W. Grosse-Kunstleve, J.-S. Jiang, J. Kuszewski, M. Nilges, N. S. Pannu, R. J. Read, L. M. Rice, T. Simonson, and G. L. Warren. Crystallography and NMR system (CNS): A new software suite for macromolecular structure determination. *Acta Cryst. sect. D*, 54:905–921, 1998.

[35] T. K. Mal, S. J. Matthews, H. Kovacs, I. D. Campbell, and J. Boyd. Some NMR experiments and a structure determination employing a $\{^{15}\mathrm{N}, ^{2}\mathrm{H}\}$ enriched protein. *J. Biomol. NMR*, 12:259–276, 1998.

[36] P. Selenko, R. Sprangers, G. Stier, D. Buehler, U. Fischer, and M. Sattler. SMN Tudor domain structure and its interaction with the Sm proteins. *Nature Struct. Biol.*, 8(1):27–31, 2001.

[37] M. E. Noble, A. Musacchio, M. Saraste, S. A. Courtneidge, and R. K. Wierenga. Crystal structure of the SH3 domain in human Fyn; comparison of the three-dimensional structures of SH3 domains in tyrosine kinases and spectrin. *EMBO J.*, 12:2617–2624, 1993.

[38] M. Habeck, W. Rieping, and M. Nilges. Weighting of experimental evidence in macromolecular structure determination. *Proc. Natl. Acad. Sci. USA*, 103:1756–1761, 2006.

Protein structure and function prediction by pairwise comparison of hidden Markov models

Johannes Söding
Max-Planck Institute for Developmental Biology, Tübingen

Abstract

Sequence similarity search methods that identify related proteins in large sequence databases are the most important application of bioinformatics in the biological sciences, since they allow to make predictions about a protein's function, structure, and evolution
I have developed the method HHsearch for the detection of remotely related proteins, which is three times more sensitive than standard methods like PSI-BLAST and considerably faster and more sensitive than the best alternative methods. To make the method accessible to a wider community, a web server based on HHsearch was set up (`hhpred.tuebingen.mpg.de`). It can search all popular protein family databases and returns ranked matches similar to PSI-BLAST in a matter of minutes. Several options assist in the functional analysis and 3D structure prediction. This server is complemented by a related server (`hhrep.tuebingen.mpg.de`) dedicated to the detection of internal repeats in protein sequences. Using HHrep, a clear sequence signal for the structural repeats in a number of common protein folds has been detected for the first time.

1 Introduction

The class of bioinformatic tools most often used by biologists are sequence similarity search methods, of which FASTA [1], BLAST [2], and PSI-BLAST

[3] are the most popular ones (with 7782, 20288, and 16411 citations to date). These methods identify *homologous* (i.e. related) proteins in sequence databases. In cases where an experimentally characterized, homologous protein can be identified, one can make inferences about the unknown protein, because closely related proteins (e.g. with $> 50\%$ identical amino acids in the pairwise sequence alignment) generally have the same or very similar functions. But for many proteins, no significant relationship to a protein of known function can be established, especially in the most interesting cases where the protein belongs to a family that has not yet been studied.

It is still not well known among biologists that, when conventional sequence search methods fail, recently developed, highly sensitive methods for remote homology detection [4, 5, 6] or structure prediction [7, 8, 9, 10, 11, 12, 13] quite often allow to make inferences from more distant relationships [14, 15, 16]. If the relationship is so remote that no common function can be assumed (e.g. if less than $\sim 30\%$ of amino acids are identical) one can generally still derive hypotheses about possible mechanisms, binding sites, functional residues, or the class of substrate bound [17].

When a homologous protein with known structure can be identified, it can be used as a *template* to model the 3D structure for the query protein [7], since even remotely homologous proteins generally have quite similar 3D structures [18]. The 3D model may then help to generate hypotheses to guide experiments.

2 Sequence alignments, sequence profiles, and HMMs

Sequence similarity search methods like FASTA or BLAST compare the sequence of a query protein with sequences of database proteins by *aligning* the two sequences, one below the other, in such a way that similar amino acids will preferably be in the same column. A *substitution matrix*, derived from the statistical analysis of many representative sequence alignments, quantifies the similarities between the twenty amino acids. The sequence similarity score is calculated as a sum over the substitution matrix elements of the aligned pairs of amino acid residues, minus penalties for gaps in the alignment. Clever heuristics speed up the calculation by a factor 10 to 100 with negligible loss in sensitivity.

The development of *profile*-to-sequence comparison methods such as PSI-BLAST [3] has lead to a vast improvement in sensitivity over these sequence–sequence comparison methods. A *sequence profile* is built from a multiple alignment of homologous sequences. It is a $20 \times L$ matrix that contains the fraction of each of the twenty amino acids in each of the L columns of the multiple alignment (Figure 1). The profile can be understood as a concise

statistical representation of the sequence family alignment, which contains more information about the sequence family than a single sequence. The profile allows to distinguish between conserved positions that are important for defining members of the family and nonconserved positions that are variable among the family members. More than that, it describes exactly how likely we are to find each of the amino acids at each position, which is why sequence profiles are sometimes called "position-specific substitution matrices". In practice, profile-based search methods work in an iterative fashion. After each search round, they add the significantly related sequences to the multiple alignment from which the sequence profile for the next round of database search is constructed.

```
HBA_human  ... W  G  K  V  G  A  -  -  H  A  G  E  ...
HBB_human  ... W  G  K  V  -  -  -  -  N  V  D  E  ...
MYG_physa  ... W  G  K  V  E  A  -  -  D  V  A  G  ...
LGB2_luplu ... W  K  D  F  N  A  -  -  N  I  P  K  ...
GLB1_glydi ... W  E  E  I  A  G  A  D  N  G  A  G  ...
```

A	...	0	0	0	0	0.25	0.75		0	0.2	0.4	0	...
C	...	0	0	0	0	0	0		0	0	0	0	...
D	...	0	0	0.2	0	0	0		0.2	0	0.2	0	...
E	...	0	0.2	0.2	0	0.25	0		0	0	0	0.4	...
F	...	0	0	0	0.2	0	0		0	0	0	0	...
G	...	0	0.6	0	0	0.25	0.25		0	0.2	0.2	0.4	...
H	...	0	0	0	0	0	0		0.2	0	0	0	...
I	...	0	0	0	0.2	0	0		0	0.2	0	0	...
K	...	0	0.2	0.6	0	0	0		0	0	0	0.2	...
L	...	0	0	0	0	0	0		0	0	0	0	...
M	...	0	0	0	0	0	0		0	0	0	0	...
N	...	0	0	0	0	0.25	0		0.6	0	0	0	...
P	...	0	0	0	0	0	0		0	0	0.2	0	...
Q	...	0	0	0	0	0	0		0	0	0	0	...
R	...	0	0	0	0	0	0		0	0	0	0	...
S	...	0	0	0	0	0	0		0	0	0	0	...
T	...	0	0	0	0	0	0		0	0	0	0	...
V	...	0	0	0	0.6	0	0		0	0.4	0	0	...
W	...	1.0	0	0	0	0	0		0	0	0	0	...
Y	...	0	0	0	0	0	0		0	0	0	0	...

Fig. 1: A multiple sequence alignment and its associated sequence profile.

A significant improvement over profile–sequence based methods was made possible by comparing profiles to profiles. These methods use PSI-BLAST or a similar method to build a profile for a query sequence and compare this profile with a database of precomputed profiles. Several such programs for homology recognition have recently been developed: LAMA [4], PROF_SIM [5], and COMPASS [6]. They were shown to be significantly more sensitive than PSI-BLAST and have been applied to identify evolutionary links between protein families previously thought to be unrelated. In addition, almost all of the top structure prediction servers now rely on profile–profile com-

parison, as can be seen from the results of the blind, automated structure prediction contest CAFASP [19].

Profile hidden Markov models (HMMs) are similar to simple sequence profiles, but in addition to the amino acid frequencies they contain the position-specific probabilities for inserts and deletions along the multiple sequence alignment. The logarithms of these probabilities are in fact equivalent to position-specific gap penalties [20]. Not surprisingly, profile HMMs perform better than sequence profiles in the detection of homologous proteins and in the quality of alignments [21,22,23], but despite the success of profile-profile alignment methods, the generalization to HMM-HMM comparison has not been done until recently.

3 Pairwise alignment of HMMs

A statistical theory of pairwise alignment of HMMs was independently developed by Lyngsgø *et al.* [24] and myself [25]. Both approaches start from the *co-emission probability* as a measure of similarity of two profile HMMs, i.e. the probability that the two aligned HMMs *emit* the same amino acid at each aligned position[1]. Lyngsgø *et al.* find the alignment that maximizes the logarithm of the co-emisssion probability, whereas our method sets the co-emission probability in relation to a *null model* probability describing the probability of emitting the same sequence under the assumption of unrelated proteins. More precisely, I look for the alignment that maximizes the score, defined as the logarithm of the sum over all co-emittable sequences of the ratio of co-emission probability to null model probability. It can be shown [25] that the use of a null model considerably improves performance by giving more weight to the co-emission of rarer amino acids[2]. Furthermore, by including a null model our HMM-HMM alignment score reduces to the successful log-odds score of HMM-to-sequence alignment in the case when one HMM is constructed from a single sequence.

A profile HMM contains in each column a match state M, a delete state D and an insert state I (Figure 2). The transition probabilities between states, symbolized by the arrows, are calculated from the insert and deletion frequencies at each position in the multiple alignment. To align two HMMs, they must be able to emit the same sequence of amino acids in aligned columns.

[1] When interpreting the HMM as a generative model, we say Match states *emit* amino acids according to the amino acid distribution of the corresponding multiple alignment column. Insert states emit amino acids with a probability distribution equal to some mean frequencies in a sequence database.

[2] I speculate that limited performance due to the lack of a null model as well as a relatively cumbersome algorithm may be reasons why the method of Lyngsgø was never developed further or made publicly available.

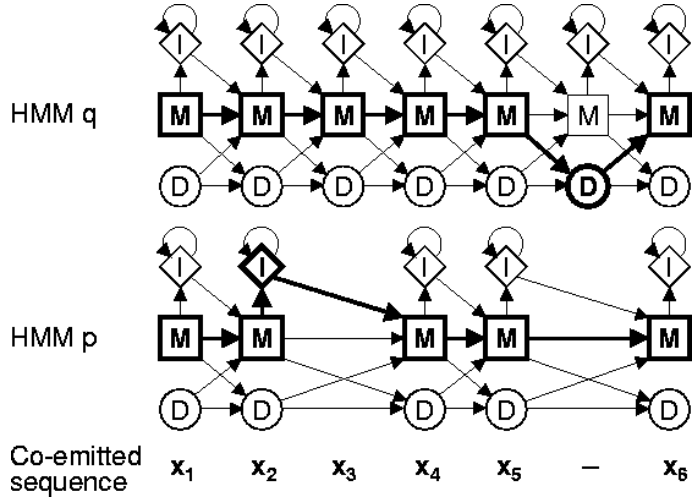


Fig. 2: Alignment of two HMMs. The path through the two HMMs corresponds to a sequence that is co-emitted by both HMMs. M: match states, D: delete states, I: insert states. Match states can emit amino acids with a probability distribution given by the corresponding column of the multiple alignment (see previous figure).

Match and insert states emit amino acids whereas delete states don't. Therefore, a match or insert state in one HMM can only be aligned with a match or insert state in the other HMM. Conversely, a delete state can only be aligned with a delete state or with nothing (i.e. with a *gap*) (Figure 2). As an example, in the third column of the alignment in Figure 2, HMM q emits a residue from its M state and HMM p emits a residue from the I state. In column six of the alignment, HMM q does not emit anything since it passes through the D state. HMM p does not emit anything either, since it has a gap in the alignment.

I use *dynamic programming* to iteratively solve the task of finding the highest-scoring alignment of two HMMs. Simply speaking, one calculates the score of the optimal alignment up to columns (i,j) from the optimal alignments up to ($i-1$,$j-1$), ($i-1$, j), and (i, $j-1$).

Since protein structures diverge much more slowly than sequences, it makes sense to include a comparison of secondary structures when aligning putative remotely homologous sequences. I have developed a statistical method akin to amino acid substitution matrices that also takes into account the confidence with which the secondary structure state (alpha helix, beta strand, or coil) at each position is predicted. The similarity scores between predicted states are then simply added to the amino acid-based scores of each column.

4 Benchmark comparison

I performed an all-against-all comparison with various similarity search tools to test their ability to detect remotely homologous proteins and to produce high-quality alignments below the twilight zone [26] of sequence similarity. I compared BLAST, PSI-BLAST, the HMM–sequence comparison package HMMER , the profile–profile alignment tools PROF_SIM and COMPASS, and our method HHsearch. In order to pinpoint the source of improvements, I benchmarked four versions of HHsearch. HHsearch 0 uses simple profile–profile comparison, HHsearch 1 is the basic HMM–HMM version, HHsearch 2 includes a novel correlation score [25], and HHsearch 3 and 4 additionally score secondary structure similarity (with predicted vs. predicted and predicted vs. actual secondary structure).

The SCOP hierarchical database [27] of structural domains was used as test set, since any pair of sequences from the same SCOP superfamily can safely be assumed to be homologous, whereas every pair with different folds are assumed to be unrelated. We will call these pairs *true positives* and *false positives*, respectively. SCOP (version 1.63) was filtered to obtain a set of 3691 sequences with a maximum pairwise *sequence identity* of 20% (i.e. having no more than 20% identical residues in a pairwise alignment). A multiple alignment was built from each sequence by using PSI-BLAST with up to eight iterations and an HMM was calculated from each of theses alignments.

Sensitivity: In order to assess the ability of the methods to distinguish true from false positives, we plot in Figure 3 the number of true positives versus the number of false positives detected above a score threshold. The ideal method would detect all homologous relationships before the first non-homologous pair is reported, yielding a vertically rising graph.

In short, HHsearch finds about twice as many homologous pairs at constant error rate of 10% (dashed diagonal line) as the next best method and more than three times as many as PSI-BLAST or HMMER. The improvement over the best alternative method (COMPASS) is due, to about one third, to the inclusion the statistical scoring scheme and the preparation of profiles (compare traces for COMPASS and HHsearch 0 in Figure 3). Another third is gained by using HMMs instead of simple profiles (compare HHsearch 0 with 1), and the last third is owed to the inclusion of secondary structure and correlation scoring (compare HHsearch 1 with HHsearch 3 or 4).

Alignment quality: In comparative structure modeling, the alignment quality between query and template sequence is the key determinant of model quality [28]. The quality of sequence alignments can be assessed by looking at the spatial distances between aligned pairs of residues upon superposition of their 3D structures. A measure for this structural fit is the MaxSub score [29]. It is 1.0 for a perfect structural fit between query and template

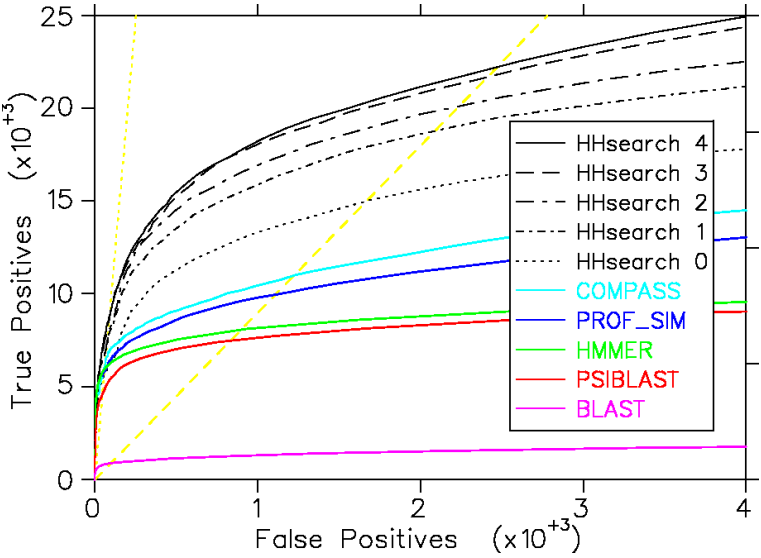


Fig. 3: Sensitivity of various homology detection tools, measured by how many true positives are detected at varying numbers of false positives. True positives are homologous pairs, false positives are unrelated. For definitions of HHsearch 0–4 please refer to the main text.

structures when all query residues are aligned at RMSD of 0 Å, and zero if the similarity is found to be insignificant.

Figure 4 plots the percentage of pairs found with scores in ten score bins, when all pairs with proteins from the same family (a) or the same superfamily (b) are considered. Pairs with MaxSub score of zero are omitted from the lowest bin. In conclusion, HHsearch is able to produce more alignments with scores above 0 than any of the other methods, and this difference increases for the more difficult inter-family alignments (b).

Speed: HHsearch scans a query sequence of 200 residues against 3691 domains in 33 s on an Athlon 64 3200+ PC. This is 10 times faster than PROF_SIM, 17 times faster than COMPASS, and only 2.5 times slower than the HMM-to-sequence comparison method HMMER. This speed was achieved through an efficient algorithm, rigorous profiling, and implementation of fast logarithm and power functions. In addition, HHsearch is parallelized (using POSIX threads) to run on multiple processors of an SMP machine, with good scaling properties (2 CPUs give a speed-up of $\times 1.7$ on a dual Athlon 64 3200+ machine under Linux).

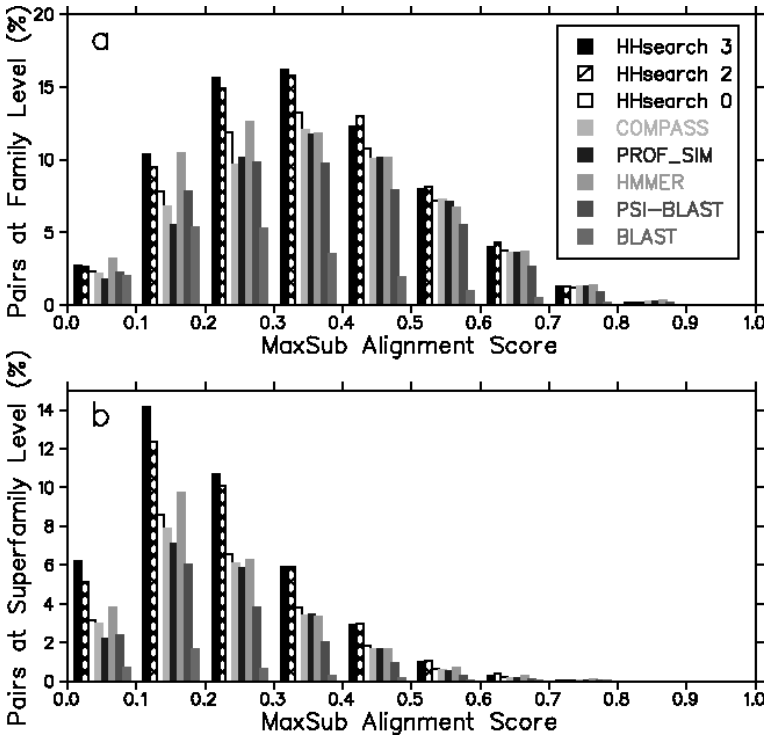


Fig. 4: Distribution of MaxSub scores for alignments of domain pairs related at the family or superfamily level in percent. Counts with MaxSub score of zero are not shown.

5 The HHpred server for structure and function prediction

The web server HHpred was developed to provide biologists with a method for sequence database searching that is as easy to use as BLAST or PSI-BLAST and yet competitive in sensitivity with the most powerful, specialized servers for structure prediction available.

Upon submission of a query sequence (or alignment), HHpred proceeds in three steps. First, an alignment of homologs is built for the query sequence by multiple iterations of PSI-BLAST searches. In the next step, a profile HMM is generated from the multiple alignment that also includes the information about predicted secondary structure. In the last step, the query HMM is compared to each HMM in the selected database. The database HMMs have been precalculated and also contain secondary structure information, either predicted by PSIPRED, or assigned from 3D structure by DSSP [30]. The server then presents the results organized into three sections: a graphical overview

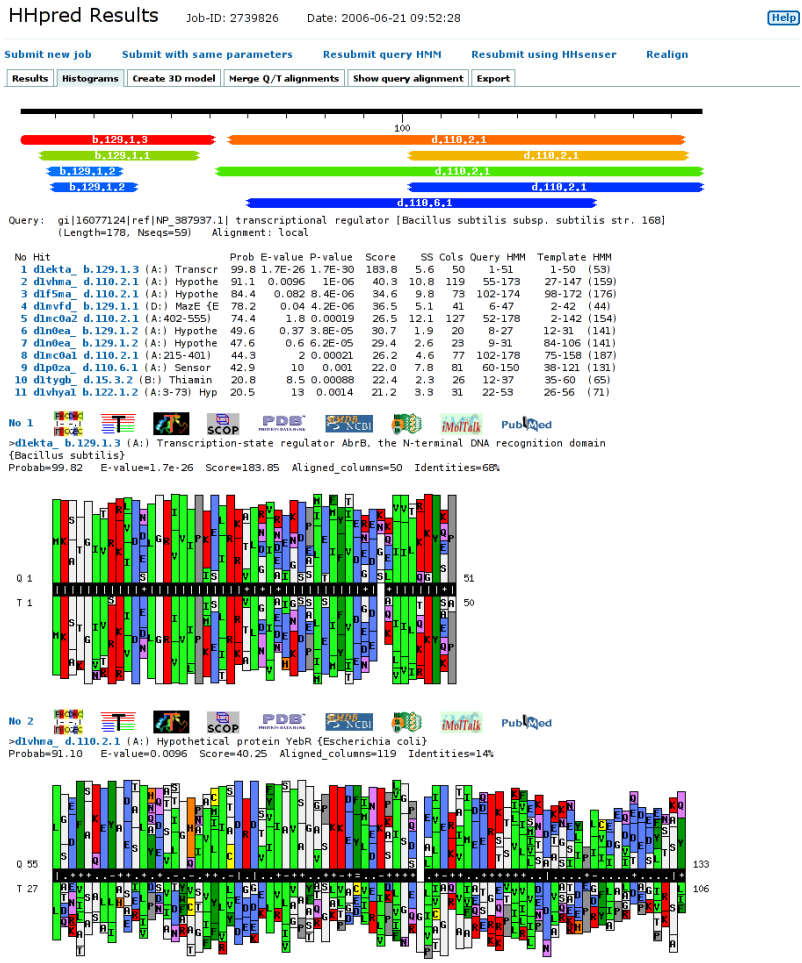


Fig. 5: Search results for HHpred at the example of transition state regulator SpoVT. The bar graph and summary hit list at the top show that SpoVT consists of two domains: the N-terminal domain is very similar to AbrB (rank 1) and clearly homologous to MazE (rank 4) and the C-terminal domain is similar to GAF and PAS domains (rank 2, 3, 5 etc). In the summary table, column 'Prob' lists the probability that the hit is homologous to the query. The alignments can be either displayed as annotated sequence alignments, or, like here, as histograms giving the amino acid distributions in the profile HMM columns. Amino acids with less than 10% are not shown. The coloring indicates the class, e.g. aliphatic, aromatic, polar etc. Various links and options provide means to further analyze results or generate a structural model.

of location and significance of the best matches, a summary table, and detailed pairwise query-template alignments (Figure 5). I believe that HHpred is unique in the advantages it offers:

Databases: In addition to the PDB and SCOP databases, all of the standard protein family databases can be searched and are automatically updated: Pfam [31], SMART [32], COG/KOG [33], CDD [34], InterPro [35], TIGRFAM [36], Panther [37], PIRSF [38], and CATH/Gene3d [39]. This sets HHpred apart from most other servers able to detect remote homologies, which are generally more specialized for protein structure prediction and only offer searches of the PDB.

User-friendliness: Search results are presented in an easy-to-read format similar to BLAST. Alignments contain annotation about secondary structure, consensus sequences, and position-specific reliability, and a histogram view of the HMM-HMM alignments permit to quickly identify functional motifs (Figure 5).

Flexibility: We try to offer the user maximum control and flexibility. One can paste one's own query alignment, search in local or global alignment mode, realign with other parameters, edit the query-template (multiple) alignment with which to launch the comparative modeling, merge the query HMM with database HMMs for *intermediate profile search*, view structures of templates or computed models, and so forth. Furthermore, HHpred is embedded in our web-based MPI Bioinformatics toolkit, which integrates many in-house and public tools in one convenient environment.

Selectivity: High-scoring false positives have systematically been reduced by developing a protocol for building query and database alignments that suppresses non-homologous sequences (J. Söding, to be published).

Sensitivity: HHpred is among the most sensitive servers for remote homology detection. A comparison of the new version with the servers that took part in the structure prediction benchmark CAFASP4 [19] can be viewed at `http://protevo.eb.tuebingen.mpg.de/hhpred/hhpred_in_CAFASP4.html`. Recently, we have integrated a new method for automatized exhaustive intermediate profile search, HHsenser [40], which can be called from within HHpred, to further increase sensitivity.

Documentation: Detailed help pages (>13000 words) are available.

6 Applications

HHpred now processes over 2000 external queries per month. A year after publication of HHsearch and HHpred, I found 36 articles citing them. Of these, three applied HHsearch on a large scale as a main method of analysis [41, 42, 43] and another 17 employed HHsearch or HHpred for detecting a remote homology relationship or predicting a 3D protein structure [44, 45,

46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]. In the following, four examples that illustrate the use of HHsearch and HHpred are presented.

6.1 Structure and function prediction for Rtv from the fruit fly

The retroactive (rtv) gene of *Drosophila melanogaster* was identified at our institute in a genetic screen for chitin-associated developmental defects. No annotated homologous sequence were found with standard methods. A manual sequence search with PSI-BLAST and submission of the resulting alignment to HHpred resulted in the detection of a family of snake toxins and a family of extracellular receptor domains as distant homologs. From the latter relationship we could conclude that Rtv is an extracellular protein anchored to the cell membrane by a GPI linkage. Modeling the structure of the Rtv protein with a set of diverse templates resulted in a structure with three long, floppy, exposed loops that are held together by five disulphide bridges. The length of the loops is unique among relatives of Rtv, and each carries two exposed, aromatic residues at the end. Exposed, aromatic residues are known to bind sugar-derivatives like chitin. This lead us to the hypothesis that Rtv is involved in binding and organizing the chitin fibers emerging from the epithelial cell surface [60]. Recent preliminary experimental evidence confirms this prediction.

6.2 Detection of tandem BRCT domains in human Nbs1

Human Nbs1 (and its homolog Xrs2 from yeast) are part of the conserved MRN complex which plays a crucial role in maintaining genomic stability. NBS1 corresponds to the gene mutated in the Nijmegen breakage syndrome known as a radiation hyper-sensitive disease. Despite the importance of the MRN complex, the high sequence divergence between Nbs1 and Xrs2 prevented the identification of common domains downstream of the N-terminal Fork-Head Associated (FHA) domain. Using HHpred, a team from Saclay could identify three as yet undetected BRCT domains in Nbs1 and Xrs2 downstream of FHA [47]. Based on the hand-refined HHpred alignments, a structural model of FHA with the two BRCT domains was built, leading to the prediction that the duplicated BRCT domain acts as phospho-serine binding module in phosphorylation-dependent protein-protein interactions and to the identification of the binding surface for the phospho-serine carrying interaction partner. The model further shows that the phospho-binding sites of FHA and BRCT are at least 45 Å apart and, surprisingly, that there is not a single residue of linker between FHA and BRCT in any of the homologs, which hints at a tight coupling between the phospho-binding functions of the FHA and BRCT domains.

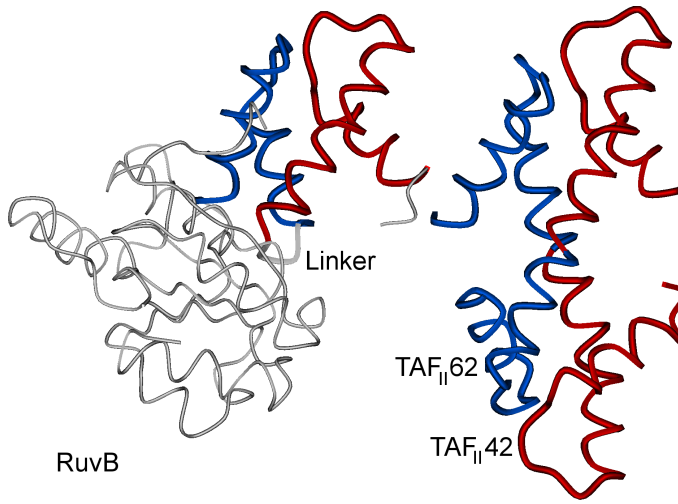


Fig. 6: The histone fold, represented here by the heterodimer TAF$_{II}$62/TAF$_{II}$42, evolved from the C-terminal subdomain of AAA+ ATPases like RuvB by the deletion of a linker and a 3D domain swap.

6.3 Novel members of the PD(D/E)XK nuclease superfamily

The PD(D/E)XK nuclease superfamily of Mg^{2+}-dependent nucleases groups together protein domains found in diverse enzymes involved in DNA replication, repair, and recombination. Typically, the sequence similarity between these proteins is so low that most members of this superfamily could be classified as PD-(D/E)XK nuclease only after their structures were determined experimentally. To find new members of this superfamily, Kosinski *et al.* [41] used the HHsearch package to build HMMs for all known members and to search the Pfam and COG databases for significant similarities. They report the identification of a PD(D/E)XK nuclease domain in numerous proteins implicated in interactions with DNA, but with unknown structure and function. The work will help to jump-start the experimental characterization of new nucleases, of which many will be important for the understanding of mechanisms that govern the evolution and stability of the genome.

6.4 Evolution of histones from a subdomain of AAA+ ATPases

In an all-against-all screen of homologous relationships between members of different folds (see section 4), we identified a striking similarity both in sequence and in structure between the histone proteins and the small helical subdomain of extended ATPase domains in AAA+ proteins (Figure 6). This

relationship is remarkable since it is conventionally assumed that different folds share no common ancestors. We conclude that the histones evolved from the ATPase subdomain, consisting of two alpha-helical hairpins connected by a short linker, by deletion of the linker and merging of the two inner helices into a long straight helix, and subsequent dimerization in order to preserve the tertiary interactions ("3D domain swap") [61].

7 The HHrep server for *de novo* repeat detection

Six out of the ten most populated folds possess an approximate structural symmetry [62, 63]. Most proteins that adopt one of these folds have no symmetry detectable in their sequences, however, and it is unclear for most domain families in these folds whether their structural symmetry has its cause in an origin through duplication. The ability to detect these structural repeats by their sequences would open a window to study hypotheses about the origin of these domains by duplication of simpler fragments. Furthermore, the detection of structural repeat patterns could help to predict the fold and function of sequences for which no detectable homolog with known structure can be found.

There are two general classes of methods to detect repeats in protein sequences. The first use their own database of profile HMMs or sequence profiles which are constructed from known repeat families, and they compare these profiles one by one with the query sequence. The second class is called *de novo* repeat detection methods: They do not rely on *a priori* knowledge about repeat families. Instead, they look for internal similarities by comparing the protein sequence to itself with standard sequence-sequence alignment techniques.

HHrep [64] is a web server for *de novo* identification of repeats in protein sequences, which is based on the pairwise comparison of HMMs. Its main strength is its sensitivity, allowing it to detect highly divergent repeat units in protein sequences whose repeats could as yet only be detected from their structures. Examples include sequences with β-propeller fold, ferredoxin-like fold, double psi barrels, or $(\beta\alpha)_8$ (TIM) barrels.

This is illustrated in Figure 7 at the example of the $(\beta\alpha)_8$ barrel structure of KDPG aldolase, by revealing a clear fourfold symmetry which we detect solely from sequence information. This symmetry points to an ancient origin through duplication of a $\beta\alpha\beta\alpha$ unit [64] and not, as previously hypothesized, by duplication of a half-barrel [65].

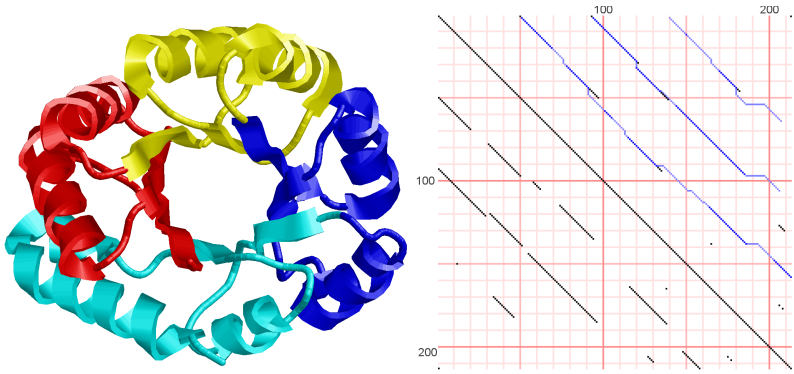


Fig. 7: The structure of $(\beta\alpha)_8$ barrels like KDPG aldolase (1fq0_A) is made up of four structural repeat units (left). The dot plot generated by HHrep shows for the first time a clear fourfold repeat structure in the sequence of a $(\beta\alpha)_8$ barrel (right). A black dot at position (i, j) indicates that profile columns i and j are similar.

8 The HHsearch software package

The HHsearch software package is programmed in C++ with a modular and object-oriented design. It consists of a program `hhmake` to generate HMMs from multiple alignments, `hhsearch` to search a database of HMMs (simple obtained by concatenating single HMMs), and `hhalign` to perform pairwise optimal or stochastically sampled alignment of HMMs and to generate dot plots as used by HHrep. In addition, a fast utility `hhfilter` can select a representative set of sequences by a maximum pairwise sequence identity criterion. With the software, users can download twelve standard, free family alignment databases in HHsearch-readable format, including our own pdb70 and scop70 databases.

Several scripts written in perl are offered with the package: `reformat.pl` can transform many standard multiple alignment formats into each other, `alignblast.pl` can parse a multiple alignment from PSI-BLAST output, `addpsipred.pl` adds predicted secondary structure to FASTA-formatted alignments or HMMs, and `hhmakemodel.pl` can parse the output of hhsearch or hhalign and generate merged multiple alignments in various formats or create rough 3D models.

The web servers have been set up as part of our MPI bioinformatics toolbox for protein sequence analysis [66] in a model-view-controller web framework. The new version is completely rewritten in Ruby on Rails (`http://www.rubyonrails.org/`) and will soon be released. It will also be made freely available under the GLPL license.

9 Summary and outlook

The HHsearch software for remote homology detection through pairwise comparison of HMMs has already found numerous applications in protein structure prediction, protein function prediction, and protein evolution, of which only a few could be mentioned in section 6. The HHpred web server was developed to make this method accessible to a wider community and to greatly enhance its functionality and usability for structure and function prediction. The HHrep server, which is based on the same method for HMM-HMM comparison, represents the most sensitive tool for *de novo* repeat detection in proteins.

I envisage many developments for protein function and structure prediction that build on the present methods. (1) A planned extension to HHpred is a PDBalert system. Users can enter a list of proteins in a web form, which will be automatically checked every week for similarity with the newly released protein structures. (2) We are working on a new method for comparative modeling that employs Bayesian statistics and advanced Markov chain Monte Carlo sampling techniques to simultaneously determine an optimal structural model and an improved query-template alignment (In collaboration with M. Habeck). (3) I will explore a way to speed up HHsearch by a factor of 10–100 by condensing the information contained in a single profile column into a discrete state alphabet and using fast heuristics developed for sequence-sequence comparison [1, 2] to pre-screen for potential homologs. This could enable HHsearch to reach the speed of PSI-BLAST at much higher sensitivity.

Acknowledgements

I would like to thank Andreas Biegert and Michael Remmert for their invaluable help in setting up the web servers. I am indebted to Andrei Lupas for his advice in the design of the servers, as well as for his constant support. Last, I thank all users who gave us feedback to improve our software.

References

[1] Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U. S. A.,* **85**, 2444–2448.

[2] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool. *J Mol Biol,* **215**, 403–410.

[3] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acid Research,* **25**, 3389–3402.

[4] Pietrokovski, S. (1996) Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Res.,* **24**, 3836–3845.

[5] Yona, G. and Levitt, M. (2002) Within the twilight zone: a sensitive profile–profile comparison tool based on information theory. *J. Mol. Biol.,* **315**, 1257–1275.

[6] Sadreyev, R. I. and Grishin, N. V. (2003) COMPASS: A tool for comparison of multiple protein alignments with assessment of statistical significance. *J. Mol. Biol.,* **326**, 317–336.

[7] Rychlewski, L., Zhang, B., and Godzik, A. (1998) Fold and function predictions for Mycoplasma genitalium proteins. *Fold Des,* **3**, 229–238.
[8] vonÖhsen, N., Sommer, I., and Zimmer, R. (2003) Profile–profile alignment: a powerful tool for protein structure prediction. *Pac. Symp. Biocomput.,* pp. 252–263.
[9] Panchenko, A. R. (2003) Finding weak similarities between proteins by sequence profile comparison. *Nucleic Acids Res.,* **31**, 683–689.
[10] Fischer, D. (2003) 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins,* **51**, 434–441.
[11] Ginalski, K., Pas, J., Wyrwicz, L. S., vonGrotthus, M., Bujnicki, J. M., and Rychlewski, L. (2003) ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acid Research,* **31**, 3804–3807.
[12] Ginalski, K., Elofsson, A., Fischer, D., and Rychlewski, L. (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics,* **19**, 1015–1018.
[13] Tang, C. L., Xie, L., Koh, I. Y., Posy, S., Alexov, E., and Honig, B. (2003) On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *J. Mol. Biol.,* **334**, 1043–1062.
[14] Ginalski, K., Rychlewski, L., Baker, D., and Grishin, N. V. (2004) Protein structure prediction for the male-specific region of the human Y chromosome. *Proc Natl Acad Sci USA,* **101**, 2305–2310.
[15] Pawlak, S. D., Radlinska, M., Chmiel, A. A., Bujnicki, J. M., and Skowronek, K. J. (2005) Inference of relationships in the 'twilight zone' of homology using a combination of bioinformatics and site-directed mutagenesis: a case study of restriction endonucleases Bsp6I and PvuII. *Nucleic Acids Res,* **33**, 661–671.
[16] Kihara, D. and Skolnick, J. (2004) Microbial genomes have over 72% structure assignment by the threading algorithm PROSPECTOR_Q. *Proteins,* **55**, 464–473.
[17] Todd, A. E., Orengo, C. A., and Thornton, J. M. (2001) Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.,* **307**, 1113–1143.
[18] Kinch, L. and Grishin, N. (2002) Evolution of protein structures and functions. *Curr. Opin. Struct. Biol.,* **12**, 400–408.
[19] Fischer, D., Rychlewski, L., Dunbrack, R. L. J., Ortiz, A. R., and Elofsson, A. (2003) CAFASP3: the third critical assessment of fully automated structure prediction methods. *Proteins,* **53**, 503–516.
[20] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998) Biological sequence analysis: Probabilistic models of proteins and nucleic acids, Cambridge University Press, Cambridge, .
[21] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994) Hidden markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.,* **235**, 1501–1531.
[22] Eddy, S. R. (1998) Profile hidden markov models. *Bioinformatics,* **14**, 755–763.
[23] Karplus, K., Karchin, R., Barrett, C., Tu, S., Cline, M., Diekhans, M., Grate, L., Casper, J., and Hughey, R. (2001) What is the value added by human intervention in protein structure prediction. *Proteins,* **45**, 86–91.
[24] Lyngsø, R. B., Pedersen, C. N. S., and Nielsen, H. (1999) Metrics and similarity measures for hidden markov models. *Proc. Int. Conf. Intell. Syst. Mol. Biol.,* pp. 178–186.
[25] Söding, J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics,* **21**, 951–960.
[26] Doolittle, R. F. (1981) Similar amino acid sequences: chance or common ancestry. *Science,* **214**, 149–159.
[27] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.,* **247**, 536–540.
[28] Venclovas, C. (2003) Comparative modeling in CASP5: progress is evident, but alignment errors remain a significant hindrance. *Proteins,* **53**, 380–388.

[29] Siew, N., Elofsson, A., Rychlewski, L., and Fischer, D. (2000) MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics,* **16**, 776–85.

[30] Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers,* **22**, 2577–2637.

[31] Sonnhammer, E. L., Eddy, S. R., Birney, E., Bateman, A., and Durbin, R. (1998) Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res,* **26**, 320–322.

[32] Ponting, C. P., Schultz, J., Milpetz, F., and Bork, P. (1999) SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nucleic Acids Res,* **27**, 229–232.

[33] Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., and Natale, D. A. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics,* **4**, 41–41.

[34] Marchler-Bauer, A., Panchenko, A., Shoemaker, B., Thiessen, P., Geer, L., and Bryant, S. (2002) CDD: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Res.,* **30**, 281–283.

[35] Mulder, N. J., Apweiler, R., and /it et al., A. (2005) InterPro, progress and status in 2005. *Nucleic. Acids. Res.,* **33**, D201–D205.

[36] Haft, D. H., Selengut, J. D., and White, O. (2003) The TIGRFAMs database of protein families. *Nucleic. Acids. Res.,* **31**, 371–373.

[37] Mi, H., Lazareva-Ulitsky, B., Loo, R., Kejariwal, A., Vandergriff, J., Rabkin, S., Guo, N., Muruganujan, A., Doremieux, O., Campbell, M. J., Kitano, H., and Thomas, P. D. (2005) The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic. Acids. Res.,* **33**, D284–D288.

[38] Wu, C. H., Nikolskaya, A., and Huang, H. *et al.*. (2004) PIRSF: family classification system at the Protein Information Resource. *Nucleic. Acids. Res.,* **32**, D112–D114.

[39] Pearl, F., Todd, A., and Sillitoe, I. *et al.*. (2005) The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic. Acids. Res.,* **33**, D247–D251.

[40] Söding, J., Remmert, M., Biegert, A., and Lupas, A. N. (2006) HHsenser: detection of remotely homologous protein sequences by intermediate profile search and HMM-HMM comparison. *Nucleic Acids Res.,* **34**, in press.

[41] Kosinski, J., Feder, M., and Bujnicki, J. M. (2005) The PD-(D/E)XK superfamily revisited: identification of new members among proteins involved in DNA metabolism and functional predictions for domains of (hitherto) unknown function. *BMC Bioinformatics,* **6**, 172.

[42] Suhre, K. (2005) Gene and genome duplication in Acanthamoeba polyphaga Mimivirus. *J. Virology,* **79**, 14095–14101.

[43] Liu, J., Glazko, G., and Mushegian, A. (2006) Protein repertoire of double-stranded DNA bacteriophages. *Virus Res.,* **117**, 68–80.

[44] Ammelburg, M., Frickey, T., and Lupas, A. N. (2006) Classification of AAA+ proteins. *J. Struc. Biol.,* in press.

[45] Djuranovic, S., Rockel, B., Lupas, A. N., and Martin, J. (2006) Characterization of AMA, a new AAA protein from Archaeoglobus and methanogenic archaea. *J. Struc. Biol.,* in press.

[46] Diemand, A. and Lupas, A. N. (2006) Modeling AAA+ ring complexes from monomeric structures. *J. Struc. Biol.,* in press.

[47] Becker, E., Meyer, V., Madaoui, H., and Guerois, R. (2006) Detection of a tandem BRCT in Nbs1 and Xrs2 with functional implications in the DNA damage response. *Bioinformatics,* **22**, 1289–1292.

[48] Tkaczuk, K. L., Obarska, A., and Bujnicki, J. M. (2006) Molecular phylogenetics and comparative modeling of HEN1, a methyltransferase involved in plant microRNA biogenesis. *BMC Evo. Biol.,* **6**, 6.

[49] Boekhorst, J., Helmer, Q., Kleerebezem, M., and Siezen, R. J. (2006) Comparative analysis of proteins with a mucus-binding domain found exclusively in lactic acid bacteria. *Microbiology-SGM,* **152**, 273–280.

[50] Albrecht, R., Zeth, K., Söding, J., Lupas, A. N., and Linke, D. (2006) Expression, crystallization and preliminary X-ray crystallographic studies of the outer membrane protein OmpW from Escherichia coli. *Acta. Crystallograph. Sect. F. Struct. Biol. Cryst. Commun.,* **62**, 415–418.

[51] Devos, D., Dokudovskaya, S., Williams, R., Alber, F., Eswar, N., Chait, B. T., Rout, M. P., and Sali, A. (2006) Simple fold composition and modular architecture of the nuclear pore complex. *PNAS,* **103**, 2172–2177.

[52] Dokudovskaya, S., Williams, R., Devos, D., Sali, A., Chait, B. T., and Rout, M. P. (2006) Protease accessibility laddering: a proteomic tool for probing protein structure. *Structure,* **14**, 653–660.

[53] Neugebauer, H., Herrmann, C., Kammer, W., Schwarz, G., Nordheim, A., and Braun, V. (2005) ExbBD-dependent transport of maltodextrins through the novel MalA protein across the outer membrane of Caulobacter crescentus. *J. Bact.,* **187**, 8300–8311.

[54] Minakhin, L., Semenova, E., Liu, J., Vasilov, A., Severinova, E., Gabisonia, T., Inman, R., Mushegian, A., and Severinov, K. (2005) Genome sequence and gene expression of Bacillus anthracis bacteriophage Fah. *J. Mol. Biol.,* **354**, 1–15.

[55] Gibson, A., Lewis, A. P., Affleck, K., Aitken, A. J., Meldrum, E., and Thompson, N. (2005) hCLCA1 and mCLCA3 are secreted non-integral membrane proteins and therefore are not ion channels. *J. Biol. Chem.,* **280**, 27205–27212.

[56] Jin, J., Cai, Y., Yao, T., Gottschalk, A. J., Florens, L., Swanson, S. K., Gutierrez, J. L., Coleman, M. K., Workman, J. L., Mushegian, A., Washburn, M. P., Conaway, R. C., and Conaway, J. W. (2005) A mammalian chromatin remodeling complex with similarities to the yeast INO80 complex. *J. Biol. Chem.,* **280**, 41207–41212.

[57] Tilburn, J., Sanchez-Ferrero, J. C., Reoyo, E., Arst, H. N., and Penalva, M. A. (2005) Mutational analysis of the pH signal transduction component PalC of Aspergillus nidulans supports distant similarity to BRO1 domain family members. *Genetics,* **171**, 393–401.

[58] Chatterjee, I., Richmond, A., Putiri, E., Shakes, D. C., and Singson, A. (2005) The Caenorhabditis elegans spe-38 gene encodes a novel four-pass integral membrane protein required for sperm function at fertilization. *Development,* **132**, 2795–2808.

[59] Coles, M., Djuranovic, S., Söding, J., Frickey, T., Koretke, K., Truffault, V., Martin, J., and Lupas, A. N. (2005) AbrB-like transcription factors assume a swapped hairpin fold that is evolutionarily related to double-psi beta barrels. *Structure,* **13**, 919–928.

[60] Moussian, B., Söding, J., Schwarz, H., and Nüsslein-Volhard, C. (2005) Retroactive, a membrane-anchored extracellular protein related to vertebrate snake neurotoxin-like proteins, is required for cuticle organization in the larva of Drosophila melanogaster. *Dev. Dyn.,* **233**, 1056–1063.

[61] Alva Kullanja, V., Ammelburg, M., Söding, J., and Lupas, A. N. (2006) The origin of the histone fold. In preparation.

[62] Salem, G. M., Hutchinson, E. G., Orengo, C. A., and Thornton, J. M. (1999) Correlation of observed fold frequency with the occurrence of local structural motifs. *J. Mol. Biol.,* **287**, 969–981.

[63] Söding, J. and Lupas, A. N. (2003) More than the sum of their parts: on the evolution of proteins from peptides. *Bioessays,* **25**, 837–846.

[64] Söding, J., Remmert, M., and Biegert (2006) HHrep: *de novo* protein repeat detection and the origin of TIM barrels. *Nucleic Acids Res.,* **34**, in press.

[65] Lang, D., Thoma, R., Henn-Sax, M., Sterner, R., and Wilmanns, M. (2000) Structural evidence for evolution of the β/α barrel scaffold by gene duplication and fusion. *Science,* **289**, 1546–1550.

[66] Biegert, A., Remmert, M., Söding, J., and Lupas, A. N. (2006) The MPI Bioinformatics Toolkit for protein sequence analysis. *Nucleic Acids Res.,* **34**, in press.

Weitere Beiträge für den Heinz-Billing-Preis 2006

A Design Tool for Modeling Asynchronous Dynamic Logic

Frank Sill, Frank Grassert
Claas Cornelius, Dirk Timmermann
Institute of Applied Microelectronics & Computer Engineering
University of Rostock, Germany

Abstract

For high performance designs, dynamic logic techniques have to be considered due to the promising high reachable frequencies. Such a technique is the True Single Phase Clock (TSPC) logic that allows designing circuits with standard cells and high speed potential. However, the disadvantages are a difficult clock tree design and high power consumption. Asynchronous logic has the potential to solve these problems. The used technique in this work, Asynchronous Chain-TSPC logic, assembles small asynchronous chains of dynamic logic gates into one period of the global clock. The results are shorter latency for calculations, power reduction due to reduced overall input load and due to no need for latches as well as a simpler clock distribution network with increased clock skew tolerance and reduced clock load.

Current high level synthesis tools do not support automated synthesis and verification of asynchronous dynamic logic. Thus, this contribution presents a complete design flow for Asynchronous Chain-TSPC logic. We use the toolset DYNAMIC, which realizes a transformation of a combinational circuit into a pipelined structure and the tool AC-DYNAMIC which implements the conversion of a pipelined structure into an asynchronously clocked structure. Furthermore, AC-DYNAMIC is capable of verifying the timing behavior and undertakes optimizations. The design flow is exemplarily applied for a 32-bit-single-error-correcting circuit.

Introduction

The exponential increase in performance and functionality has been the key to success for the microelectronics industry. However, in the past years power dissipation has reached alarming levels and has become the limiting factor for future performance enhancements. Gordon Moore, originator of the well-known Moore's law, pointed out this correlation at the worldwide largest conference on research and applications in microelectronics, the International Solid State Circuits Conference (ISSCC). He demanded extensive research efforts for reducing power dissipation in order to maintain the exponential performance increase of electronic systems in the forthcoming decade as well (Moore, 2003).

Asynchronous techniques are a promising, but also challenging, approach to cut down on power consumption. An example for such a technique is the asynchronous circuit style AC-TSPC (Asynchronous Chain True Single Phase Clock) which was developed at the University of Rostock (Grassert, 2005). AC-TSPC allows a reduction in energy dissipation of high performance processors to approximately one third. Generally, asynchronous approaches do not rely on a global clock signal to synchronize the logic elements which implement the system's functionality. Instead, AC-TSPC's logic elements begin to evaluate after receiving a start signal from its predecessors. Accordingly, every logic element creates a start signal after finishing its computation.

This type of signaling is most problematic for common circuit simulators which are specifically developed to handle synchronous designs (Sill, 2002). Thus, it is not possible with these simulators to verify the timing behavior and the evaluation results of asynchronous circuits. So, the absence of such automated design tools and design flows is one of the reasons why asynchronous concepts did not become widely accepted in spite of their advantages.

In this work the simulation and design tool AC-DYNAMIC is presented which tackles exactly this problem. With the help of complex algorithms for analysis, this tool models the timing behavior of each gate and examines the evaluation of all appearing signal combinations. Thereby, for each signal a time frame can be defined in which the signal changes its state. This approach allows the additional consideration of process based variations of signal propagation delay which will have a tremendous impact in future technologies (Srivastava, 2005). Based on the simulation results the designer can finally determine the exact behavior of the AC-TSPC design and he can verify the signaling between all deployed gates. Furthermore, the designer receives recommendations to optimize the design in terms of performance.

The modeling is not limited to be used for AC-TSPC only so that the implemented algorithms can be used for other asynchronous design

techniques as well which highly facilitates the development of these techniques because the verification is one of the main problems in chip design (Weste, 2005). Furthermore, AC-DYNAMIC is integrated in the standard design flow and, thus, does not require additional adjusted or modified design steps.

The next section describes the fundamentals of AC-TSPC's self-timed structure and the important completion detection with its problems in larger designs. The functionality of the developed simulation and design tool, which includes high level timing calculation based on the timing values of basic gates, is discussed in the following section before the results from transistor level simulations are presented and final conclusions of this work are drawn.

Self-timed structures

This section introduces the basic principles of self-timed structures. Therefore, differential dynamic logic is briefly explained that allows to easily detect completion of evaluation. Additionally, self-timed schemes in principle and the implemented asynchronous technique AC-TSPC are presented.

Dynamic logic and the differential usage

The functionality of dynamic logic is depicted in figure 1 and compared to static CMOS (SCMOS), the most widespread circuit technique. The logic function of a dynamic gate is implemented solely with a network of either N-MOS or P-MOS transistors while the logic function in SCMOS is implemented with both N-MOS and P-MOS transistors in a complementary setup. Consequently, dynamic logic is faster, because of the smaller number of transistors that contribute to the capacitive input load. In the given example of figure 1 only two transistors are needed compared to four in SCMOS. Furthermore, N-MOS transistors have a smaller input capacitance than P-MOS transistors when sized with equal driving strength.

Dynamic logic works in two phases, which are dictated by a clock signal Φ. The following explanation is valid for an N-logic block but can similarly be understood for P-logic as well. During precharge phase (clock low), the P-MOS transistor connects the output node Y of the dynamic gate to $V_{DD,}$ thus, charging the output capacitance at node Y to high. In the evaluation phase (clock high) the clocked N-MOS transistor is turned on while the P-MOS transistor is turned off. Depending on the input values of the logic tree (X1 and X2 in figure 1) the output node is possibly discharged to

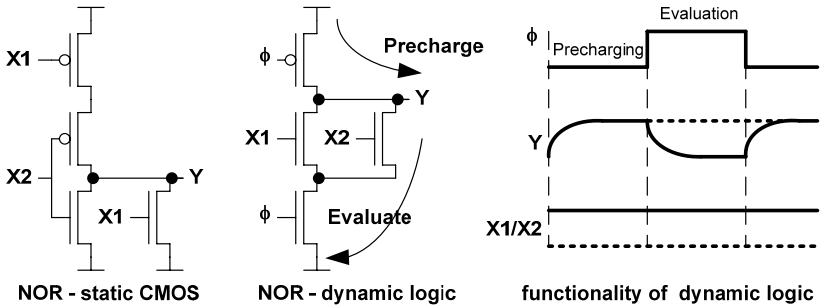


Fig. 1: NOR gate implemented in static CMOS and dynamic logic as presented together with the signaling behavior of dynamic logic

ground. Therefore, it is the evaluation phase that realizes the underlying logic function. A P-logic block works in a complementary manner with the clock signal being high during precharge and vice versa.

TSPC logic (see figure 2) uses alternating dynamic N-logic and P-logic blocks combined with N- or P-latches, respectively. Thereby, N- and P-blocks evaluate and precharge both in one clock cycle. The speed of the system is determined by the block with slowest action, i.e. the evaluation of a logic-block together with its latch (P or N) or the precharge of internal nodes (to high or low; V_{DD} or ground, respectively).

Another representative of dynamic circuit techniques is the DOMINO logic that uses N-logic blocks only with subsequent static inverting logic, e.g. an inverter. This is required to allow cascading several gates sequentially which will be explained in the following. Consider the case that two simple dynamic gates (as shown in figure 1) are cascaded and use the same clock signal. The outputs are charged high during precharge so that the second gate will temporarily start to discharge its output at the beginning of the evaluation phase. This results in signal degradation and eventually in malfunction. By using a static inverter, as shown in figure 2, each output of such a stage goes low during precharge. Therefore, subsequent logic trees can not connect to ground.

Another way to allow cascading dynamic gates is to use gates with different clocks so that a gate accepts the output signal of a preceding gate only if both evaluation phases overlap. In doing so, the evaluation phase of the first gate has to hold until the internal node of the second gate is fully settled. Otherwise, information is lost. Though, the shifted precharge of the first gate does not affect the settled outputs of the next one because of solely high to low transitions at the inputs.

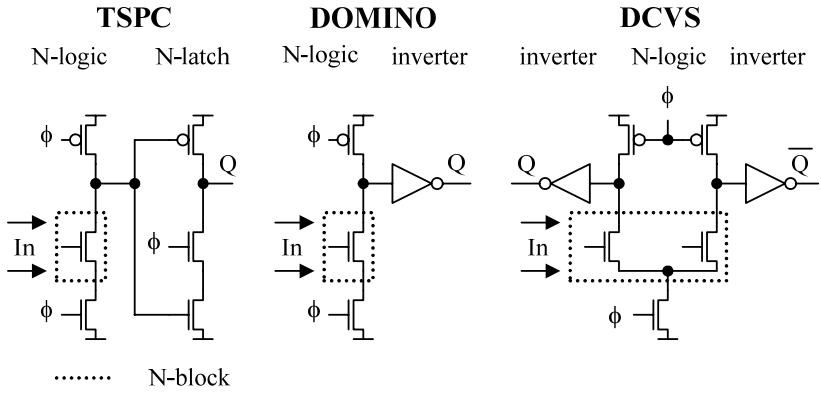


Fig. 2: Examples of three common dynamic circuit techniques: TSPC, DOMINO and dynamic DCVS

The given examples allow deriving another circuit technique with differential outputs. The structure can be understood as two DOMINO gates with complementary logic trees and a shared clocked N-MOS transistor. An example of such an approach, the Differential Cascode Voltage Switch (DCVS) logic, is given in figure 2. The mode of operation is similar to DOMINO but the differential logic structure always evaluates two complementary output values. This allows using the differential outputs at the end of evaluation as completion detection. It should be mentioned that the same functionality can be realized if two independent domino stages are used instead.

Basic Self-timed Scheme

Dynamic logic with a complementary structure and differential outputs can be arranged in an asynchronous way. There are two main approaches to set up a self-timed scheme for dynamic logic: gate outputs control the clocking of the previous or the clocking of the following gate (Krambeck, 1982). The first structure is advantageous and enables a simple implementation with minimum evaluation time. If the evaluated outputs of a gate have settled, a completion signal is generated which sets the previous gate back into the precharge phase because the inputs have successfully been processed – start precharge.

Similarly, completely precharged outputs set the previous gate into the evaluation phase (i. e. the own outputs have been precharged – start next evaluation at the predecessor; new inputs can be processed). Because the evaluation starts only with valid inputs, every gate is waiting for valid input signals during the evaluation phase. Therefore, evaluation time is only the sum of gate evaluation times with no extra delays. The computations start with the first gate and propagate the chain without being additionally delayed. Figure 3 shows such a self-timed structure with dynamic dual-rail gates. Here, a NOR gate is applied to generate the completion signal that can directly be used as the clock signal for the previous gate when a dual rail structure with DOMINO (Harris, 1997), (Yee, 2000) or DCVSL (Heller, 1984) is used.

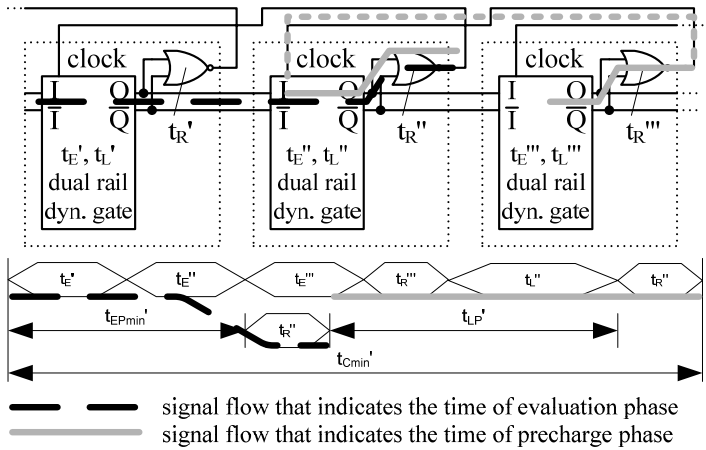


Fig. 3: Dual-rail self-timed structure and timing chart of the precharge and evaluation phase of the first logic level

AC-TSPC used in this work

The particular characteristic of Asynchronous Chain True Single Phase Clock (AC-TSPC) logic is that asynchronous chains with self-timed structures are aligned by a global clock signal (see figure 4, left hand side). To integrate such chains of logic in a synchronous design, a single phase global clock with the same duration of high and low phases clocks the last gate of each chain (Grassert, 2002). Starting from there, all previous gates are connected via the self-timed scheme.

If the runtime of the chain is nearly half the clock cycle time, the evaluation will be delayed just before the last gate. However, this does not corrupt the functionality because the resulting structure of AC-TSPC still behaves like separated pipeline stages which are controlled by the global clock signal. The gates within the asynchronous chains (i. e. the pipeline stages) are classified in vertical so called chain stages where the outputs of the gates must connect only to gates in the following stage. As a gate can be connected to more than one gate with its outputs, this gate can be a part of different chains. To cope with this, the naming convention of virtual chains is established (see figure 4, right hand side). Such virtual chains represent all possible combinations of a gate within the different existing chains and are required in the simulator and design tool for the determination of timing behavior of the circuit.

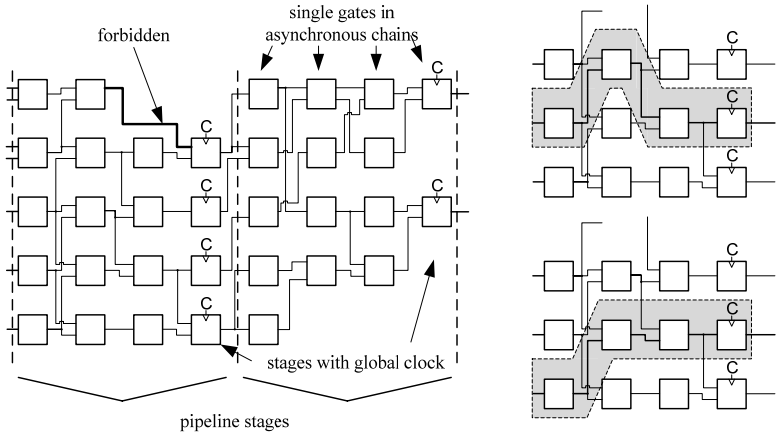


Fig. 4: Pipelined structure of AC-TSPC with globally clocked stages; Virtual chains representing different signal paths

Developed Design Flow

This section introduces a synthesis strategy for implementing complex designs using the self-timed logic structures which were developed and presented in "Dynamic Single Phase Logic with Self-timed Stages for Power Reduction in Pipeline Circuit Designs" (Grassert, 2001). To date, the ability to apply AC-TSPC structures to large and complex designs has been restricted by the missing automated timing analysis. The developed tool AC-DYNAMIC does not just allow such simulations but also supports

further necessary steps in an automated design flow. We present a method to combine a standard Synopsys design flow with a SCMOS library and the developed library for the self-timed structures. Thereto, three additional steps after the standard synthesis are required before the actual simulation can take place. Five steps are required in total whereas the first one only needs to be performed once for all subsequent designs:

- Compiling of a new Synopsys synthesis library
- Synthesis with Synopsys design compiler
- Micro Pipeline Reorganization (MPR) done with DYNAMIC
- Setup of chain structure with AC-DYNAMIC
- Optimization performed with AC-DYNAMIC

This resulting design flow is depicted in figure 5 and the given steps are introduced in a more detailed manner in the following.

Development of a design library

The development of a design library is a prerequisite before the actual design flow can be used. All necessary gates and logic functions have to be included in a new Synopsys (Synopsys, 2006) synthesis library. *HSPICE* was used for the gate level simulations before the library compiler was applied to create the library containing the results in terms of timing values, structure and logical function. Additionally, we collected the values for the minimum and maximum time of evaluation, precharge and generation of the self-timed signals which are to be used in AC-DYNAMIC for the timing analysis. The library itself has to be composed such that for every static gate in the standard library, a functionally equivalent dynamic dual-rail gate exists. Furthermore, the dynamic dual-rail gates with completion detection were designed.

Automated design of dynamic pipeline stages

Automatic synthesis of dynamic pipeline stages requires a couple of additional steps. In the first place, a basic VHDL description is developed that includes the functionality in an abstract way. Then, the synthesis of the abstract design description is carried out with the design compiler and the developed library of static gates. At this point, a netlist with combinational logic only exists, i. e. without any clocking signals (see figure 6, left hand side).

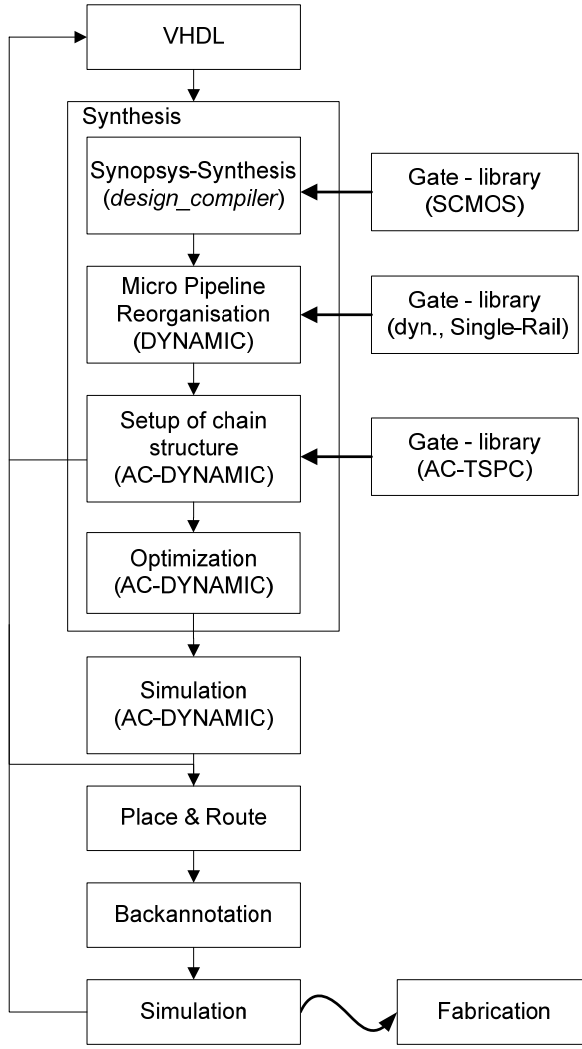


Fig. 5: Adapted standard design flow for the generation of integrated circuits with AC-TSPC logic

This netlist has to be processed with the Micro Pipeline Reorganization (MPR) tool DYNAMIC (Wassatsch, 2002). Because every dynamic gate includes a register function, the tool can handle each logic gate as a pipeline stage. It replaces the combinational gates with equivalent single-rail dynamic gates. Then it includes simple registers in single wires to ensure correct timing behavior, because each signal has to be registered in every clock cycle. The required approach for the conversion of static gates into a pipelined structure is illustrated by a simple example in figure 6 (right hand side). The tool DYNAMIC finally generates a netlist for the fully pipelined logic. The condition for the use of the pipeline tool is the existence of a combinational netlist without any recursive connections. This means that no combinational feedback may exist, because the tool needs a well defined starting point and ending point for each signal path (refers to the virtual chains defined earlier). Where a combinational loop can be split into a forward part and a feedback part, the forward part can also be processed with the DYNAMIC tool.

Setup of asynchronous chains

The tool AC-DYNAMIC reads the netlist of the pipelined design and creates the AC-TSPC structure (see figure 4). Foremost, the dynamic single-rail gates are displaced by functional equivalent dynamic dual-rail gates with completion detection.

In the next step all input gates of the circuit are connected to the global clock signal before the last gates of all chains are connected with the global clock signal as well. Afterwards, the local completion detection signals are connected to the control inputs within the asynchronous chains. Due to irregular and complex interconnections of gates, the generation of self-timed signals used as clock signals for previous gates is not trivial. To ensure the correct order of events in the self-timed scheme, a clock signal must arrive in a defined period of time. For interconnected gates, the time periods of all participating gates must be respected. There are two main

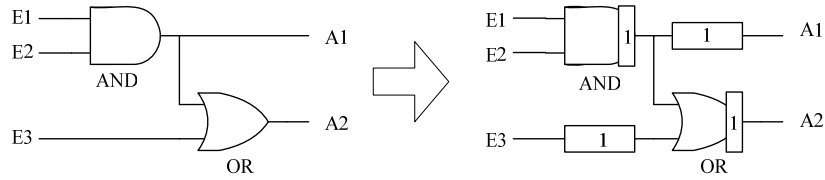


Fig. 6: Simple example for the conversion of static gates into a fully pipelined structure with equal path length

cases that have to be distinguished.

Firstly, a single completion detection signal is used as clock signal for two or more previous gates (see figure 7, left hand side). This restricts the timing behavior of the gates in the chain stage before the gate which generate the completion detection signal. The timing behavior of all previous gates, which are connected to the self-timed signal, must nearly be the same. Secondly, there are two or more gates which are connected jointly to the output signal of a single gate (see figure 7, right hand side). In this case, there is more than one completion signal that can be connected to the clock input. These signals must mostly be combined with additional logic, e.g. an OR gate. This logic has to ensure that the gate which generates the output signal does not change to precharge phase until all following gates have completed their evaluation and have also generated their completion signals. The consideration of these two cases is also included in the developed tool and appropriate actions are taken, e. g. additional logic is inserted.

Optimization of AC-TSPC

The optimization goals are low latency, high maximum frequency and little area. For this purpose, the tool AC-DYNAMIC can vary four parameters. These are the length of the chains, the completion detection logic and the length of the low and high phase of the clock signal. The tool calculates reference values for every configuration which is tested. At first, the maximum clock frequency for every possible chain length and for a symmetric or asymmetric clock signal is evaluated. The timing values of all gates and the given design constraints are used for this step. In the next step,

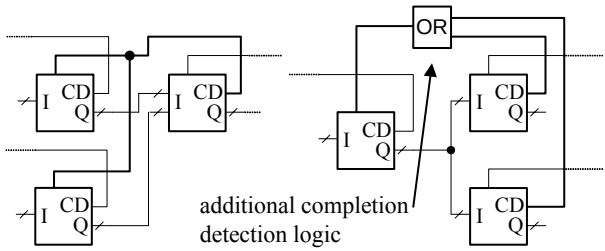


Fig. 7: Problems of completion detection due to interconnections between consecutive gates. Each block represents a dynamic dual rail gate with completion detection as shown in figure 3.

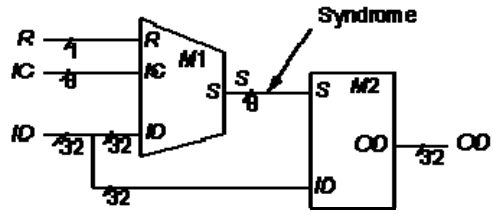


Fig. 8: Test design from the ISCAS-85 benchmark suite (C499/C1355: 32-Bit Single-Error-Correcting Circuit)

the logic for completion detection is optimized. Finally, the tool searches for chains which consist of buffers only. These chains can be replaced by a single TSPC buffer gate, because the outputs of this gate have the same behavior as the outputs of a buffered chain. Thus power and area can be preserved.

Simulation

After completing the optimization in terms of chain length, clock frequency and completion detection logic, the accurate timing behavior of the circuit is simulated by the tool AC-DYNAMIC. Because the exact evaluation and precharge timing values of a single gate depend on input vectors, temperature and process parameters, a minimum and a maximum evaluation or precharge time can be specified for each gate. The tool determines for each gate in a chain these minimum and maximum timing values and verifies that the self-timed signals do not violate these limits. To simulate the functional behavior, we use *HSPICE*.

Example

As an example we use the c1355 design from the ISCAS-85 benchmark suite which is a 32-bit single-error-correcting circuit (Hansen, 1999). The original c1355 circuit implemented in SCMOS has 41 inputs, 32 outputs, and 546 gates. The 41 inputs are combined to form an 8-bit internal bus S, which is then combined with 32 primary inputs to form the 32 primary outputs (see figure 8). The results in table 1 show that the static implementation has lower area (15% of the AC-TSPC implementation). But the maximum clock frequency of the AC-TSPC version of the circuit is

almost three times higher than the one of the SCMOS version. It must be considered, that the values for the static version are measured without flip-flops. To prove the correct functionality of the simulator and to verify the achieved timing values, transistor level simulations were performed that validated the achieved results.

Conclusions and Recommendations

This contribution presents a solution for a fully supported design flow of AC-TSPC logic. This logic style combines latch-free evaluation with fast dynamic logic to reach maximum throughput in high performance applications. The presented tool generates netlists consisting of dynamic gates, calculates the exact timing behavior using a new developed strategy and compares these values with the functional limits given by the self-timed scheme. Therefore, this tool overcomes the lack of a missing synthesis flow and exact timing analysis. The results of the extensive work on timing calculation and on implementation are the very first expressive statements of the applicability of AC-TSPC. The simulation results of the tool were validated with transistor level simulations.

Tab. 1: Comparison of results for different implementations of the ISCAS-85 c1355 benchmark design

	SCMOS	AC-TSPC
Gates	546	2098 - (AC-TSPC) 184 - (logic for completion detection)
Max. Clock frequency	193 MHz	528 MHz
Latency	5.15 ns	7.57 ns

Acknowledgment

The authors acknowledge the SNUG technical committee member Kurt Baty for his assistance and helpful suggestions. Parts of this work were supported by the German Research Foundation (DFG) under grant number GRK-466 and the VIVA project.

References

Grassert, F. and Timmermann, D., “Dynamic Single Phase Logic with Self-timed Stages for Power Reduction in Pipeline Circuit Designs”, IEEE International Symposium on Circuits and Systems (ISCAS), May 2001.

Grassert, F. and Timmermann, D., “Asynchronous Chain True Single Phase Clock Logik (ACTSPC)”, 3. Schwerpunktkolloquium des DFG Schwerpunktprogramms Grundlagen und Verfahren verlustarmer Informationsverarbeitung (VIVA), ISBN: 3-00-008995-0, p.136 - 141, Chemnitz, March 2002.

Grassert, F., Verlustleistungsoptimierte Schaltungstechniken fuer hoechste Geschwindigkeiten, phd. thesis, Rostock, Germany, 2005.

Hansen, Yalcin, M. H., and Hayes, J. P., "Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering," IEEE Design and Test, vol. 16, no. 3, pp. 72-80, July-Sept. 1999.

Harris, D. and Horowitz, M. A., “Skew-Tolerant Domino Circuits”, IEEE Journal of Solid-State Circuits, Vol. 32, No. 11, Nov. 1997.

Heller, L. G., Griffin, W. R., Davis, J. W., and Thoma, N. G., “Cascode Voltage Switch Logic: A Differential CMOS Logic Family”, Proceedings of International Solid-State Circuits Conference, IEEE, 1984, pp. 16-17.

Krambeck, R. H., Lee, C. M., and Law, H.-F. S., “High-Speed Compact Circuits with CMOS”, IEEE Journal of Solid-State Circuits, IEEE, Vol. SC-17, No. 3, Jun. 1982.

Moore, G., “No Exponential Is Forever: But ’Forever’ Can Be Delayed!”, Invited Speech, ISSCC, San Francisco, 2003.

Sill, F., “Methoden zur Verlustleistungsabschaetzung“, study, Rostock, Germany, 2002.

Srivastava, A., Sylvester, D., and Blaauw, D.: “Statistical Analysis and Optimization for VLSI: Timing and Power“, 1. Edition, Springer, 2005.

Synopsys, Inc., 700 East Middlefield Road, CA 94043-4022 United States of America. Synopsys Synthesis and Simulation Tools, 2006 edition.

Wassatsch, A. Integration dynamischer Schaltungstechnik in einen Standard-CMOS-Design-Flow mit Anwendung in der digitalen Signalverarbeitung, phd. thesis, University of Rostock, Germany, 2002.

Weste, N. H. E., and Harris, D. “CMOS VLSI Design: A Circuits and Systems Perspective“, 3. Edition, Addison-Wesley, 2005.

Yee, G. and Sechen, C., “Clock-Delayed Domino for Dynamic Circuit Design”, IEEE Transactions on VLSI Systems, Vol. 8, No. 4 , Aug. 2000.

Fingerprint-based Similarity Search and its Applications

Benno Stein
Sven Meyer zu Eissen
Bauhaus-Universität Weimar

Abstract

This paper introduces a new technology and tools from the field of text-based information retrieval. The authors have developed

- a fingerprint-based method for a highly efficient near similarity search, and
- an application of this method to identify plagiarized passages in large document collections.

The contribution of our work is twofold. Firstly, it is a search technology that enables a new quality for the comparative analysis of complex and large scientific texts. Secondly, this technology gives rise to a new class of tools for plagiarism analysis, since the comparison of entire books becomes computationally feasible.

The paper is organized as follows. Section 1 gives an introduction to plagiarism delicts and related detection methods, Section 2 outlines the method of fuzzy-fingerprints as a means for near similarity search, and Section 3 shows our methods in action: It gives examples for near similarity search as well as plagiarism detection and discusses results from a comprehensive performance analyses.

1 Plagiarism Analysis

Plagiarism is the act of claiming to be the author of material that someone else actually wrote (Encyclopædia Britannica 2005), and, with the ubiquitousness

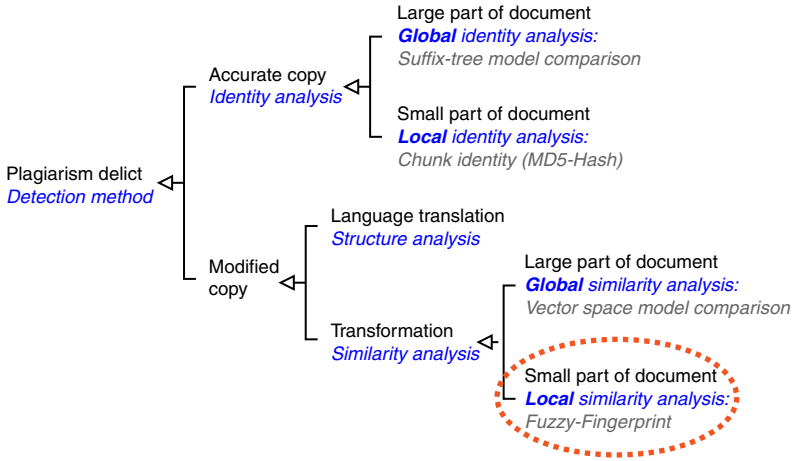


Fig. 1: A taxonomy of plagiarism delicts and analysis methods (Stein & Meyer zu Eißen 2006). The encircled part indicates the most common delict, which can be discovered with fuzzy-fingerprints.

of the World Wide Web it became more common (McCabe 2005). Plagiarism in text documents occurs in several forms: passages are copied one-to-one, passages are modified to a greater or lesser extent, or they are even translated. Clearly, a question of central importance is whether the detection of such and similar delicts can be automated. Figure 1, which is taken from (Stein & Meyer zu Eißen 2006), shows a taxonomy of plagiarism delicts along with possible detection methods. The by far most common plagiarism delict is the extraction of small parts of other authors' documents and their use in a more or less modified form within the own text (shown encircled).

Several techniques for plagiarism analysis have been proposed in the past—most of them rely on one of the following ideas:

Substring Matching. Substring matching approaches try to identify maximal matches in pairs of strings (Gusfield 1997), which then are used as plagiarism indicators. Typically, the substrings are represented in suffix trees, and graph-based measures are employed to capture the fraction of the plagiarized sections (Baker 1993, Monostori, Finkel, Zaslavsky, HodÃąsz & Pataki 2002, Monostori, Zaslavsky & Schmidt 2000). However, Finkel, Zaslavsky, Monostori & Schmidt as well as Baker propose the use of text compression algorithms to identify matches (2002, 1993).

Keyword Similarity. The idea here is to extract and to weight topic-identifying keywords from a document and to compare them to the keywords of other

documents. If the similarity exceeds a threshold, the candidate documents are divided into smaller pieces, which then are compared recursively (Si, Leong & Lau 1997, Fullam & Park 2002). Note that this approach assumes that plagiarism usually happens in topically similar documents.

Exact Fingerprint Match. The documents are partitioned into term sequences, called chunks, from which digital digests are computed that form the document's fingerprint. When the digests are inserted into a hash table, collisions indicate matching sequences. For the fingerprint computation a standard hashing approach such as MD5 hashing is employed, which suffers from two severe problems: (1) it is computationally expensive, (2) a small chunk size (3-10 words) must be chosen to identify matching passages, which additionally increases the effort for fingerprint computation, fingerprint comparison, and fingerprint storage. Recent work that describes details and variants of this approach are given in (Brin, Davis & Garcia-Molina 1995, Shivakumar & Garcia-Molina 1996, Finkel, Zaslavsky, Monostori & Schmidt 2002).

Our approach of fuzzy-fingerprinting overcomes these limitations; it is ideally suited to discover copied and slightly modified passages in large document collections. The technology was firstly published in (Stein 2005) and successfully applied to plagiarism analysis in (Stein & Meyer zu Eißen 2006). However, to understand different intentions for similarity search and its application we first introduce the distinction of local and global similarity. In fact, fuzzy-fingerprints can be understood as a combination of both paradigms, where the parameter "chunk size" (substring size) controls the degree of locality.

2 Similarity Search with Fuzzy-Fingerprints

In the context of information retrieval a fingerprint $h(d)$ of a document d can be considered as a set of encoded substrings taken from d, which serve to identify d uniquely.[1] Following Hoad & Zobel, the process of creating a fingerprint comprises four areas that need consideration (2003).

1. *Substring Selection.* From the original document substrings (chunks) are extracted according to some selection strategy. Such a strategy may consider positional, frequency-based, or structural information.
2. *Substring Number.* The substring number defines the fingerprint resolution. Obviously, there is a trade-off between fingerprint quality, processing effort, and storage requirements, which must be carefully balanced. The more information of a document is encoded in the fingerprint, the more reliably a possible collision of two fingerprints can be interpreted.

[1] The term "signature" is sometimes also used in this connection.

3. *Substring Size.* The substring size defines the fingerprint granularity. A fine granularity makes a fingerprint more susceptible to false matches, while with a coarse granularity fingerprinting becomes very sensitive to changes.
4. *Substring Encoding.* The selected substrings are mapped onto integer numbers. Substring encoding establishes a hash operation where—aside from uniqueness and uniformity—also efficiency is an important issue (Ramakrishna & Zobel 1997). For this, the popular MD5 hashing algorithm is often employed (Rivest 1992).

If the main issue is similarity analysis and not unique identification, the entire document d is used during the substring formation step—i. e., the union of all chunks covers the entire document. The total set of integer numbers represents the fingerprint $h(d)$. Note that the chunks may not be of uniform length but should be formed with the analysis task in mind.

2.1 Local and Global Similarity Analysis

For two documents A and B let $h(A)$ and $h(B)$ be their fingerprints with the respective cardinalities $|h(A)|$ and $|h(B)|$. A similarity analysis between A and B that is based on $h(A)$ and $h(B)$ measures the portion of the fingerprint intersection (Finkel et al. 2002):

$$\varphi_{local}(A, B) = \frac{|h(A) \cap h(B)|}{|h(A) \cup h(B)|}$$

We call such a kind of similarity measure *local similarity* or *overlap similarity*, because it directly relates to the number of identical regions. By contrast, the vector space model along with the cosine measure does not depend on identical regions: Two documents may have a similarity of 1 without sharing any 2-gram. The vector space model along with the cosine measure assesses a global characteristic because it quantifies the term frequency of the entire document; in particular, the model neglects word order. Figure 2 contrasts the principles of local and global similarity analysis pictorially.

Basically, a fingerprint $h(d)$ of a document d is a special document model of d. In this sense, every information retrieval task that is based on a standard document model can also be operationalized with fingerprints. However, fingerprint methods are more flexible since they can be targeted specifically towards one of the following objectives:

1. compactness—with respected to the document length
2. fidelity—with respected to a local similarity analysis

It is difficult to argue whether a fingerprint should be preferred to a standard document model in order to tackle a given retrieval task. To better understand this problem of choosing an adequate document model consider again

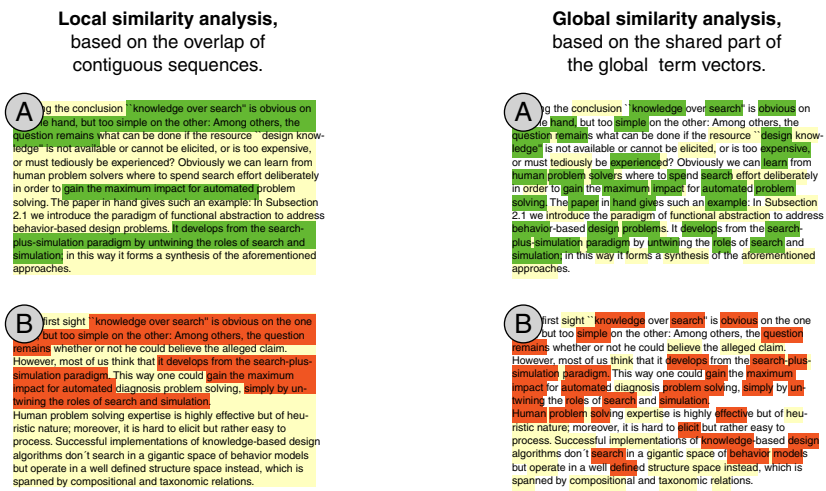


Fig. 2: Two documents A and B which are analyzed with respect to their similarity. The left-hand side illustrates a measure of local similarity: All matching contiguous sequences (chunks) with a length ≥ 5 words are highlighted. The right-hand side illustrates a measure of global similarity: Here the common word stems (without stop words) of document A and B are highlighted. Observe that both similarity analyses may lead to the same similarity assessment.

the taxonomy shown in Figure 1: here we have divided the analysis methods into local and global strategies. Note that in the literature on the subject local plagiarism analysis methods are encountered more often than global analysis methods. Among the shown approaches, the chunk identity analysis—usually operationalized with the MD5 hashing algorithm—is the most popular approach to plagiarism analysis. As already noted, such or similar methods have inherent disadvantages that can only be countered, if the chunk size is drastically increased. This, however, requires some kind of fingerprints that operationalize a "relaxed" comparison concept.

The following subsection addresses this problem. It introduces a fuzzy-fingerprint, h_φ, which is specifically tailored to text documents and which provides the desired feature: an efficient means for near similarity analysis.

2.2 *Fingerprints that Capture Near Similarity*

While most fingerprint approaches rely on the original document d, from which substrings are selected and given to a mathematical function, our approach can be developed simplest from a document's vector space model $\mathbf{d}$.

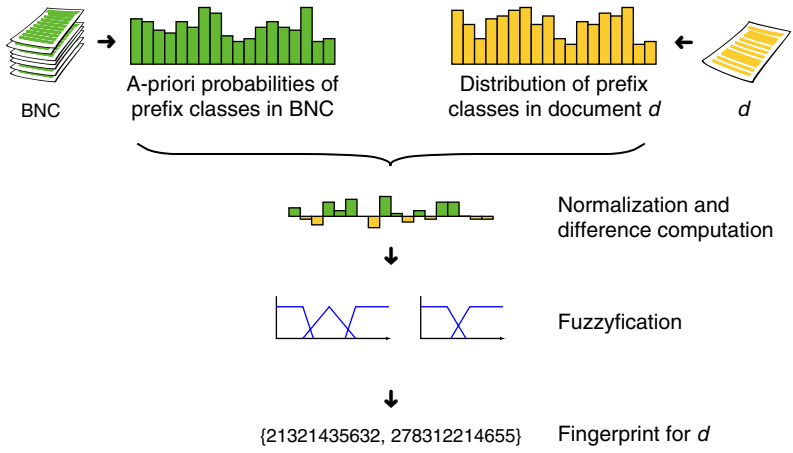


Fig. 3: Pictorial overview of the fuzzy-fingerprint construction process.

The key idea behind h_φ is an analysis and comparison of the distribution of the index terms in $\mathbf{d}$ with respect to their expected term frequency class.[2] We abstract the concept of term frequency classes towards *prefix* frequency classes, by comprising index terms into a small number of equivalence classes such that all terms from the same equivalence class start with a particular prefix. There might be the equivalence class of terms whose first character is from the set {“a”, “A”} or, as the case may be, the equivalence class of terms whose first character is from the set {“x”, “X”, “y”, “Y”, “z”, “Z”}.

Based on large corpora a standard distribution of index term frequencies can be computed and the a-priory probability of a term being member in a certain prefix class be stated. The deviation of a document’s term distribution from these a-priory probabilities forms a document-specific characteristic that can be encoded as a compact fingerprint. The following four steps define the construction of a fuzzy-fingerprint $h_\varphi(d)$ for a document $d \in D$ more precisely; Figure 3 illustrates the procedure.

1. Extraction of the set $\mathbf{d}$ of index terms from d. In connection with Web documents this includes the removal of HTML tags, scripting code, etc.

[2]The term frequency class, also called word frequency class, can be used as an indicator of a word’s customariness. Let $\mathcal{D}$ be a representative text corpus, let $|\mathcal{D}|$ be the number of words (terms) in $\mathcal{D}$, and let $f(w)$ denote the frequency of a word $w \in \mathcal{D}$. In accordance with (University of Leipzig 1995) the word frequency class $c(w)$ of a word $w \in \mathcal{D}$ is $\lfloor \log_2(f(w^*)/f(w)) \rfloor$, where w^* denotes the most frequently used word in $\mathcal{D}$. In the Sydney Morning Herald Corpus (Dennis 1995), w^* denotes the word “the”, which corresponds to the word frequency class 0; the most uncommonly used words within this corpus have a word frequency class of 19.

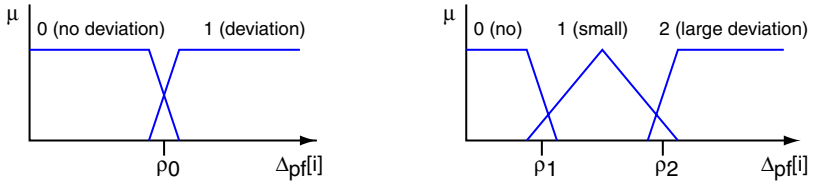


Fig. 4: The two fuzzy deviation schemes that are used for the fingerprint construction.

2. Computation of $\mathbf{pf}$, the vector of relative frequencies of k prefix classes for the index terms in $\mathbf{d}$. Our experiments rely on prefix classes that are characterized by a single alphabetic character, say, typical values for k are between 10 and 30.
3. Normalization of $\mathbf{pf}$ with respect to a reference corpus and computation of Δ_{pf}, the vector of deviations to the expected distribution. Our normalization grounds on the British National Corpus, which is a 100 million word collection of samples of written and spoken language from a wide range of sources (Aston & Burnard 1998).
4. Fuzzification of Δ_{pf} using two fuzzy deviation schemes. We propose the schemes depicted in Figure 4, which means that a deviation either falls in one of two or in one of three intervals.

Remarks. (1) The granularity of the fingerprints is controlled within two dimensions at the following places: In Step 2, by the number k of equivalence classes (= different prefix codes) to be distinguished, and in Step 4, by the resolution of the fuzzy deviation schemes. (2) Since $h_\varphi(d)$ computes a *set* of digital digests for a document d, we agree upon the following understanding of hash collisions:

$$h_\varphi(d) \cap h_\varphi(d') \neq \emptyset \;\Rightarrow\; \varphi(\mathbf{d}, \mathbf{d}') \geq 1 - \varepsilon \qquad (1)$$

(3) Finally, recall that in the vector space model all information about term order is lost. Consequently, the presented fuzzy-fingerprint approach does not encode order information either.

3 Use Cases and Performance Analysis

The purpose of this section is twofold: (1) It comprises use cases that demonstrate the wide application range of our technology, and (2) it presents analysis results that quantify different performance aspects of fuzzy-fingerprints.

3.1 Use Cases

Plagiarism in Text

The availability of educational material on the World Wide Web entices students to plagiarize from these sources. Of course, this malpractice is also observed in other situations where people can profit from plagiarized material, e. g. authors who transcribe from other papers or employees who copy

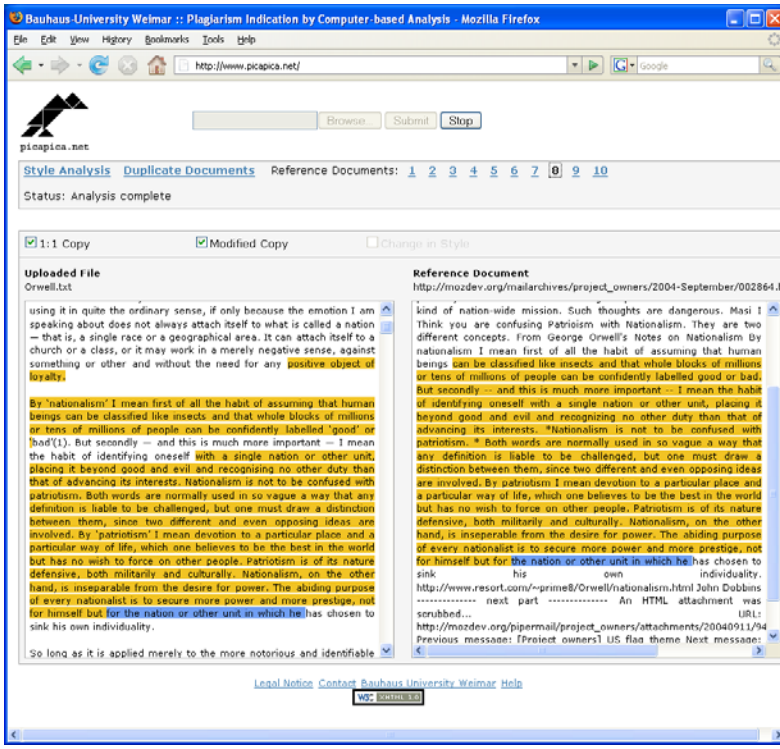


Fig. 5: The Plagiarism Finder takes an input document, automatically extracts meaningful keywords, organizes a focused Web search, and analyzes candidate documents with respect to plagiarized passages.

Fig. 6: The framed areas indicate original and plagiarized code respectively.

for their presentations. The mentioned scenarios have one aspect in common: Typically, short passages from third-party sources are slightly modified and copied into the target document. Hence they cannot be detected by traditional approaches that use cryptographic hashes. Figure 5 shows a snapshot of our Plagiarism Finder,[3] which has been developed with the fuzzy-fingerprint technology and which searches the World Wide Web for plagiarized documents.

Source Code Plagiarism

The recent lawsuit of SCO against major Linux vendors (SCO claimed that parts of Linux program sources were plagiarized from SCO Unix) shows the importance of a technology to automatically detect plagiarism at the level of program code. Since copied program source has to be adapted to fit into an existing framework, a tolerant comparison technology like fuzzy-fingerprints is necessary to identify suspicious code passages. Figure 6 shows code snippets of the same algorithm in a different context, which map onto the same fuzzy-fingerprint.

Identifying Versioned Documents

The development of a technology involves an evolution of its documentation. It is common practice to keep several versions of documents that relate to var-

[3] www.picapica.net

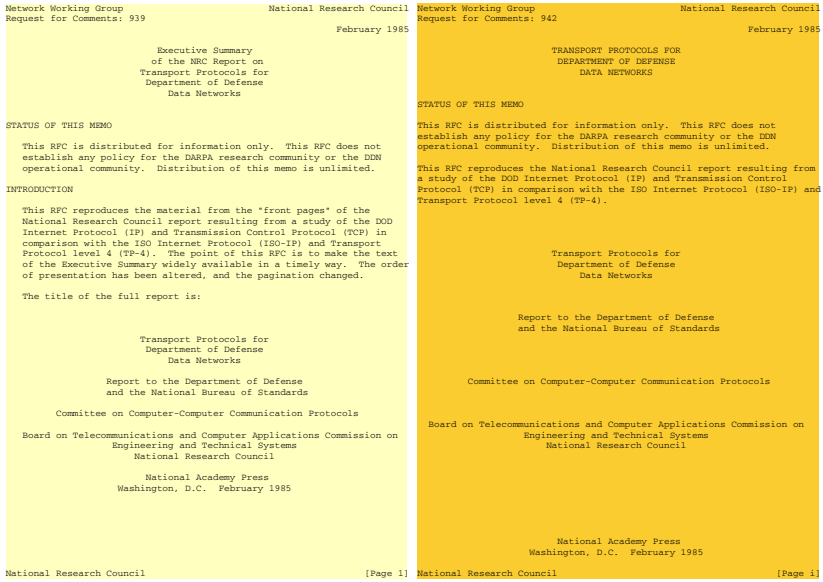


Fig. 7: Pages of two documents that represent different versions of a report about data transmission protocols. With the fingerprinting technology the respective documents are identified as variants of the same technical report.

ious software releases. Other examples for documents that develop over time include project progress reports, discussions in forums, and email threads. Fuzzy-fingerprints have proven to successfully identify versioned documents when the fingerprint is taken at the document level. Figure 7 shows an example from the RFC document collection.

Finding Similar Web Sites and Mirrors

An Internet search using the phrase “Linux Documentation Project” results in several, almost identical Web pages held at different locations, copied from each other, and revised slightly (Hoad & Zobel 2003). Of course, this phenomenon does not only apply to the mentioned project but can be observed in connection with many replicated Web pages and projects.

The identification of such sets of similar pages is useful when searching for mirror sites if a primary source is offline or overloaded. Moreover, the identification of these pages with redundant content is highly useful for search engine providers, who can optimize their storage systems using references and provide a search interface for similar Web sites. In contrast to a sequential scan of a document repository, which is expensive when the underlying

repository is huge, fuzzy-fingerprints identify similar Web sites and mirrors in constant time.

Grouping together Similar Texts

Clustering texts is the state-of-the-art methodology to identify groups of texts that share a similar subject. Fuzzy-fingerprints allow to identify groups of texts in which the pairwise similarity is above a threshold in a natural way: these groups are made up of documents that share the same fingerprint. In contrast to other methods that try to identify such groups, fuzzy-fingerprints are much faster (at least by one order of magnitude) while providing high-quality groupings.

Improvement of Text Compression

One of the most commonly used methods to compress texts is to identify common substrings and to replace them with short references. Text compression algorithms therefore hold dynamic dictionaries of frequently used substrings, which adapt according to occurrence frequency when proceeding within a text stream. Fuzzy-fingerprints can be used to group together similar texts before compressing them. This procedure allows for optimizing dictionaries for a set of texts in advance, resulting in better compression rates.

3.2 Performance Analysis

This section presents performance results of our implementation of h_φ and its application to real world similarity retrieval tasks. The purpose of our experiments is twofold. Firstly, we want to shed light on the practical performance of fuzzy-fingerprints with respect to retrieval accuracy. Secondly, we want to gain evidence on the high runtime performance of fuzzy-fingerprints in comparison with traditional approaches. The parameters of h_φ are given in Table 1.

The retrieval analysis relies on two corpora. One corpus consists of all "Request For Comments" (RFCs), a collection of about 3600 text files on In-

	Number of prefix classes k	Number of deviation intervals r	Number of fuzzification schemes ρ
h_φ	18	3	3

Tab. 1: Parameters of h_φ of the fuzzy-fingerprint. Three variations of the fuzzification scheme shown on the right-hand side in Figure 4 are used. The prefixes that define the equivalence classes are of length one, i. e. one class for each letter where the letters $\{j, k, q, u, v, x, y, z\}$ are discarded since their prefix frequencies in the English language is rather low.

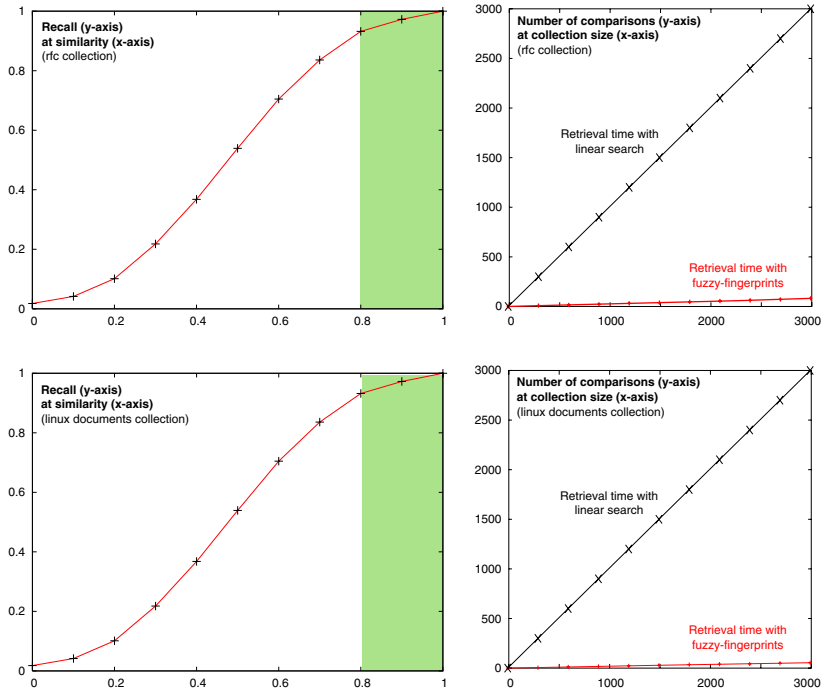


Fig. 8: The plots on the left-hand side show the recall at similarity values that were achieved for the retrieval with fuzzy-fingerprints. The plots on the right-hand side illustrate the retrieval speed up: They contrast the number of comparisons of the standard retrieval process (diagonal line) and of the fuzzy-fingerprint retrieval process.

ternet technology (Postel 2004). Since the documents in the RFC collection are versioned and thus also include updates of documents, the existence of pairs of documents with a high similarity is very likely. A second corpus is made up of about 15000 Internet documents collected with a breadth-first-search crawl starting from the "Linux Documentation Project" (Aznar 2004). For this corpus HTML documents and text documents were downloaded, the visible portion of the HTML documents were extracted, and documents with less than 50 plain words were discarded. To ensure that solely English documents are in this corpus a stopword-based language test was applied.[4] Again, documents of a high similarity are likely to occur within this collection since Linux FAQs etc. are frequently updated and, in particular, mirrored on several sites.

[4]The test is similar to the test that has been used to compile the TREC Web Collections (Text Retrieval Conference 2003).

When using fingerprints in a retrieval process instead of a "rich" document model, completeness cannot be guaranteed: There may be documents that are very similar to each other under, for instance, the vector space model—though their fingerprints are different. Hence, the key question here relates to the quality of recall, i. e., the fraction of similar documents that can be identified as such by means of their fuzzy-fingerprint.

In the experiments we employed the cosine measure along with the vector space model to assess the reference similarity, and we computed the recall values with respect to different cosine similarity thresholds. The plots on the left-hand side in Figure 8 show the resulting *recall at similarity* curves, which look very promising: For the sets of documents that are similar to each other ($> 80\%$, see the shaded area) high recall-values were achieved for queries based on the fuzzy-fingerprint retrieval.

The question of *precision* reads as follows: How many documents that yield the same fuzzy-fingerprint under h_φ are actually similar to each other? Note that the documents whose fingerprints are involved in a collision form candidates for a high similarity, and a subsequent in-depth similarity analysis based on the vector space model must be applied for them. Since with a standard retrieval approach the entire collection is investigated, the ratio between the collection size and the size of the collision set can directly be interpreted as the factor for retrieval speed-up. The plots on the right-hand side in Figure 8 illustrate the sizes of both sets: The diagonal line corresponds to the retrieval time of a sequential scan; the line below, close to the x-axis, shows the average size (and hence the retrieval time) of the collision set for fuzzy fingerprints. Obviously, the use of h_φ leads to a substantial retrieval speed-up.

References

Aston, G. & Burnard, L. (1998). The BNC Handbook, `http://www.natcorp.ox.ac.uk`.

Aznar, G. (2004). The Linux Documentation Project, `http://www.tldp.org`.

Baker, B. S. (1993). On finding duplication in strings and software, `http://cm.bell-labs.com/cm/cs/papers.html`.

Brin, S., Davis, J. & Garcia-Molina, H. (1995). Copy detection mechanisms for digital documents, *SIGMOD '95*, ACM Press, New York, NY, USA, pp. 398–409.

Dennis, S. (1995). The sydney morning herald word database, `http://www2.psy.uq.edu.au/CogPsych/Noetica/OpenForumIssue4/SMH.html`.

Encyclopædia Britannica (2005). New Frontiers in Cheating, `http://www.britannica.com/eb/article?tocId=228894`.

Finkel, R., Zaslavsky, A., Monostori, K. & Schmidt, H. (2002). Signature Extraction for Overlap Detection in Documents, *Proceedings of the 25th Australian conference on Computer science*, Australian Computer Society, Inc., pp. 59–64.

Fullam, K. & Park, J. (2002). Improvements for scalable and accurate plagiarism detection in digital documents, `http://www.lips.utexas.edu/~kfullam/pdf/DataMiningReport.pdf`.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press.

Hoad, T. & Zobel, J. (2003). Methods for Identifying Versioned and Plagiarised Documents, *American Society for Information Science and Technology* **54**(3): 203–215.

McCabe, D. (2005). Research Report of the Center for Academic Integrity, `http://www.academicintegrity.org`.

Monostori, K., Finkel, R., Zaslavsky, A., HodÃąsz, G. & Pataki, M. (2002). Comparison of overlap detection techniques, *Lecture Notes in Computer Science*, Vol. 2329, pp. 51–60.

Monostori, K., Zaslavsky, A. & Schmidt, H. (2000). Document overlap detection system for distributed digital libraries, *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, ACM Press, New York, NY, USA, pp. 226–227.

Postel, J. (2004). RFC (Request For Comments) Collection, `http://www.rfc-editor.org`.

Ramakrishna, M. & Zobel, J. (1997). Performance in Practice of String Hashing Functions, *Proceedings of the International Conference on Database Systems for Advanced Applications, Melbourne, Australia*, pp. 215–223.

Rivest, R. L. (1992). The md5 message-digest algorithm, `http://theory.lcs.mit.edu/řivest/rfc1321.txt`.

Shivakumar, N. & Garcia-Molina, H. (1996). Building a scalable and accurate copy detection mechanism, *DL '96: Proceedings of the first ACM international conference on Digital libraries*, ACM Press, New York, NY, USA, pp. 160–168.

Si, A., Leong, H. V. & Lau, R. W. H. (1997). Check: a document plagiarism detection system, *SAC '97: Proceedings of the 1997 ACM symposium on Applied computing*, ACM Press, New York, NY, USA, pp. 70–77.

Stein, B. (2005). Fuzzy-Fingerprints for Text-Based Information Retrieval, *in* K. Tochtermann & H. Maurer (eds), *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 05), Graz*, Journal of Universal Computer Science, Know-Center, pp. 572–579.

Stein, B. & Meyer zu Eißen, S. (2006). Near Similarity Search and Plagiarism Analysis, *in* M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nürnberger & W. Gaul (eds), *From Data and Information Analysis to Knowledge Engineering*, Springer, pp. 430–437.

Text Retrieval Conference (2003). The TREC Web Document Collection, `http://trec.nist.gov`.

University of Leipzig (1995). Wortschatz, `http://wortschatz.uni-leipzig.de`.

Algorithms and computer algebra from particle physics

Stefan Weinzierl
Institut für Physik, Universität Mainz

Abstract

I report on two programs, "gTybalt" and "nestedsums", which were originally developed for the calculation of quantum loop corrections in particle physics. However, these programs solve tasks, which are not specific to particle physics.

The first one, "gTybalt", is a free computer algebra system based entirely on open-source code. The program is build on top of other packages. It offers the possibility of interactive symbolic calculations within the C++ programming language. Mathematical formulae are visualized using TeX fonts.

The second package, "nestedsums", is a program library containing algorithms which allow the expansion of higher transcendental functions. These algorithms were developed by ourselves and used to solve otherwise unknown loop integrals. Applications of these algorithms extend to other branches of science, for example to number theory in mathematics.

1 Introduction

Symbolic calculations, carried out by computer algebra systems, have become an integral part in the daily work of scientists. The advance in algorithms and computer technology has led to remarkable progress in several areas of natural sciences. A striking example is provided by the tremendous progress in the last few years for analytic calculations of so-called loop diagrams in perturbative quantum field theory. It is worth to analyse what the

particular requirements on computer algebra systems for these calculations are: First of all, these tend to be "long" calculations, e.g. the system needs to process large amounts of data and efficiency in performance is a priority. Secondly, the algorithms for the solution of the problem are usually developed and implemented by the physicists themselves. This requires support from the computer algebra system for a programming language which allows to implement complex algorithms for abstract mathematical entities. In other words, it requires support of object oriented programming techniques from the system. On the other hand, these calculations usually do not require that the computer algebra system provides sophisticated tools for all branches of mathematics. Thirdly, despite the fact that these calculations process large amounts of data, the time needed for the implementation of the algorithms usually outweighs the actual running time of the program. Therefore convenient development tools are also important.

In this context I present two programs: The library "nestedsums" offers algorithms for the expansion of transcendental functions in a small parameter [1]. This expansion is the "tough" part when computing loop integrals in quantum field theory. On the other hand it is a well defined problem which can be stated in mathematics without any reference to physics. This library is an example for algorithms which are developed and implemented for a specific problem, but which are general enough to have applications to other fields.

The program "gTybalt" is a computer algebra system, which offers the possibility of interactive symbolic calculations within the C++ programming language [2]. It is a convenient development tool.

This article is organized as follows: In the next chapter I give an overview on the computer algebra system "gTybalt". In section 3 the library "nestedsums" is presented. Applications are discussed in section 4. The design of the program "gTybalt" is discussed in an appendix. Both programs are available from `http://wwwthep.physik.uni-mainz.de/~stefanw`.

2 An overview on gTybalt

gTybalt is a free computer algebra system and distributed under the terms and conditions of the GNU General Public Licence. The main features of gTybalt are:

- Object Oriented: gTybalt allows symbolic calculations within the C++ programming language.
- Efficiency for large scale problems: Solutions developed with gTybalt can be compiled with a C++ compiler and executed independently of gTybalt. This is particular important for computer-extensive problems and a major weakness of commercial computer algebra systems.

– Short development cycle: gTybalt can interpret C++ and execute C++ scripts. Solutions can be developed quickly for small-scale problems, either interactively or through scripts, and once debugged, the solutions can be compiled and scaled up to large-scale problems.
– High quality output: Mathematical formulae are visualized using TeX fonts and can easily be converted to LaTeX on a what-you-see-is-what-you-get basis.

gTybalt does not try to cover every domain of mathematics. Some desirable algorithms, like symbolic integration are not implemented. However, the modular design of gTybalt allows to incorporate easily new algorithms.

The functionality of a computer algebra system can be divided into different modules, e.g. there will be a module, which displays the output, a second module analysis and interprets the input, a third module does the actual symbolic calculation. Writing a computer algebra system from scratch is a formidable task. Fortunately it is not required, since there are already freely available packages for specific tasks. gTybalt is based on several other packages and provides the necessary communication mechanisms among these packages. gTybalt is therefore a prototype of a "bazaar"-style program and an example of what is possible within the free software community. It should be clearly stated, that without these already existing packages gTybalt would never have been developed and my thanks go to the authors of these packages for sharing their programs with others. In particular gTybalt is build on the following packages:

– The TeXmacs-editor [3] is used to display the output of formulae in high quality mathematical typesetting using TeX fonts.
– gTybalt can also be run from a text window. Then the library eqascii [4] is used to render formulae readable in text mode.
– Any interactive program needs an interpreter for its commands. gTybalt uses the CINT C/C++ interpreter [5], which allows execution of C++ scripts and C++ command line input.
– At the core of any computer algebra system is the module for symbolic and algebraic manipulations. This functionality is provided by the GiNaC-library [6].
– One aspect of computer algebra systems is arbitrary precision arithmetic. Here GiNaC (and therefore gTybalt) relies on the Class Library for Numbers (CLN library) [7].
– Plotting functions is very helpful to quickly visualize results. The graphic abilities of gTybalt are due to the Root-package [8].
– The GNU scientific library is used for Monte Carlo integration [9].
– Optionally gTybalt can be compiled with support for the expansion of transcendental functions. This requires the nestedsums library [1] to be installed.

– Optionally gTybalt can be compiled with support for factorization of polynomials. This requires the NTL library [10] to be installed.

gTybalt can be run in the TeXmacs mode or in a simple text mode. To start gTybalt in text mode, one types `gtybalt`. To quit, one types `quit`. To use gTybalt within TeXmacs, one first starts TeXmacs with the command `texmacs`. One can then start a gTybalt session by clicking on the terminal symbol and selecting "gTybalt" from the pop-up menu. Alternatively one can start gTybalt from the "Text" menu via "Text → Session → GTybalt".

2.1 Command line input

One can type in regular C++ statements which will be processed by CINT. For example

```
gTybalt> int i=1;
gTybalt> i++;
gTybalt> cout << "The increased number : " << i << endl;
The increased number : 2
```

The functionality of gTybalt for symbolic and algebraic calculations is provided by the GiNaC-library. The syntax follows the one for the GiNaC-library. For example:

```
gTybalt> symbol a("a"), b("b");
gTybalt> ex e1=pow(a+b,2);
gTybalt> print(e1);

     2
(b+a)

gTybalt> ex e2=expand(e1);
gTybalt> print(e2);

 2    2
a  + b  + 2 a b
```

Here `print` is a gTybalt-subroutine, which prints a variable to the screen. By default, gTybalt does not print anything onto the screen, unless the user specifically asks for a variable to be printed. If gTybalt is running under TeXmacs, the output will be with TeX fonts. There is also a function `rawprint` which prints the variable `e2` as follows:

```
gTybalt> rawprint(e2);
a^2+b^2+2*a*b
```

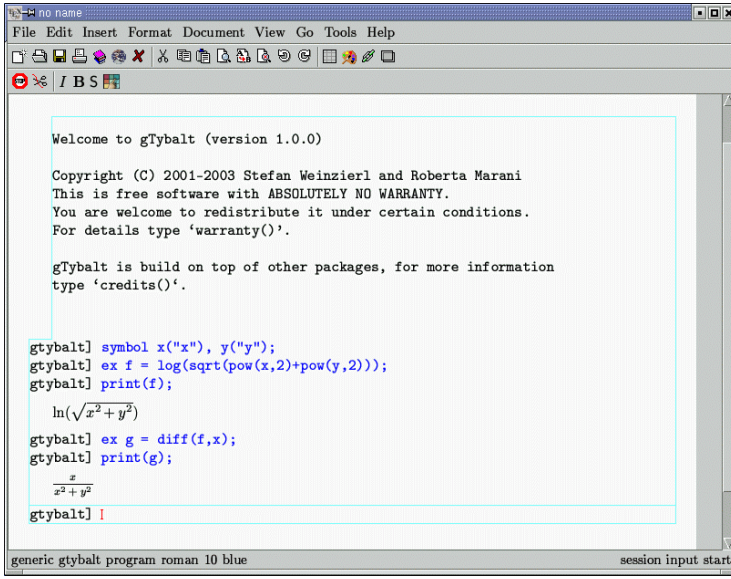


Fig. 1: A screen-shot for gTybalt when running in TeXmacs mode.

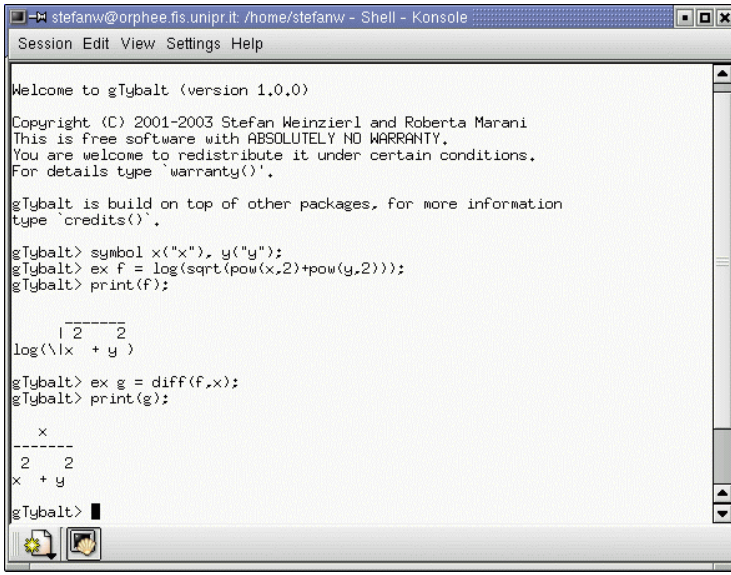


Fig. 2: A screen-shot for gTybalt when running in text mode.

Fig. 1 shows how the output of a further example will look like under TeXmacs. Fig. 2 shows the corresponding output, when gTybalt runs in text mode. Within TeXmacs mode there is the possibility to print a session to a postscript file by choosing from the "File" menu the combination "File → Export → Postscript". It is also possible to generate for a session a corresponding LaTeX file via "File → Export → LaTeX". This is in particular useful if one would like to obtain for a displayed formula the corresponding LaTeX code.

2.2 Scripts

The standard behaviour of the C++/C interpreter CINT is to interpret any command immediately. There is also the possibility to put a few commands into a script and to load this file into a session. This is done through the following commands:

```
.L file.C
.x file.C
```

The `.L` command loads a script into the session, but does not execute the script. This is useful for a script containing the definition of a function. The `.x` command loads and executes a script. As an example consider that the file `hermite.C` contains the following code:

```
ex HermitePoly(const symbol & x, int n)
{
 ex HKer=exp(-pow(x,2));
 return normal(pow(numeric(-1),n) * diff(HKer,x,n)/HKer);
}
```

This is just a function which calculates the n-th Hermite polynomial. Now try the following lines in gTybalt:

```
gTybalt> .L hermite.C
gTybalt> symbol z("z");
gTybalt> ex e1=HermitePoly(z,3);
gTybalt> print(e1);

             3
 - 12 z + 8 z
```

This prints out the third Hermite polynomial.

2.3 Plots

A function can be plotted as follows:

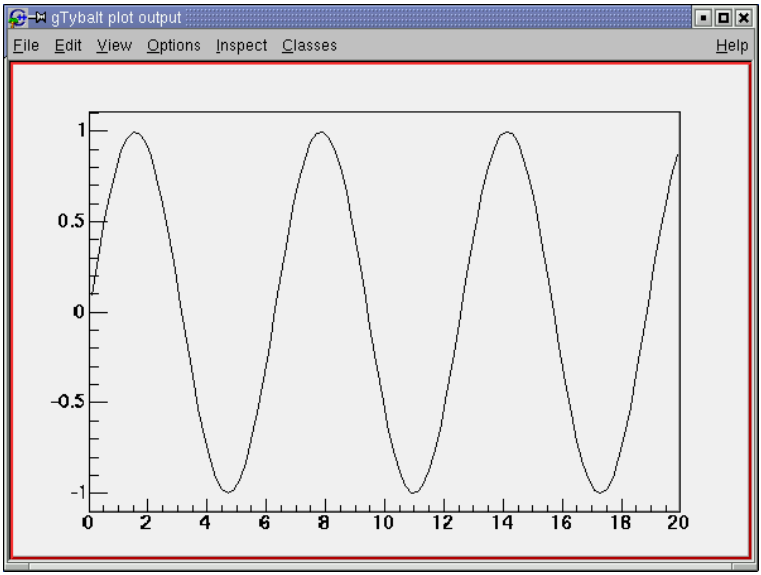


Fig. 3: The plot for the function $\sin(x)$ *for x from* 0 *to* 20.

```
gTybalt> symbol x("x");
gTybalt> ex f1=sin(x);
gTybalt> plot(f1,x,0,20);
```

This will plot $\sin(x)$ in the interval from 0 to 20. To clear the window with the plot, choose from the menu-bar of the plot "File → Quit ROOT".
Similar, a scalar function of two variables can be plotted as follows:

```
gTybalt> symbol x("x"), y("y");
gTybalt> ex f2=sin(x)*sin(y);
gTybalt> plot(f2,x,y,0,10,0,20);
```

This will plot $\sin(x)\sin(y)$ for x from 0 to 10 and y from 0 to 20. Fig. 4 shows the output from the plotting routine. To view the plot from a different angle, just grab the plot with the mouse and move it around. There is a wide variety of options on how to draw a graph. To access the draw panel, click on the right mouse button, when the mouse is placed inside the window containing the plot and choose "DrawPanel" from the pop-up menu. The options include among others lego- and contour-plots. The default option corresponds to the style "surf" and draws a (coloured) surface.
The plot can be saved to a file. For example, to save the plot as a postscript file, choose from the "File" menu the option "Save As canvas.ps".

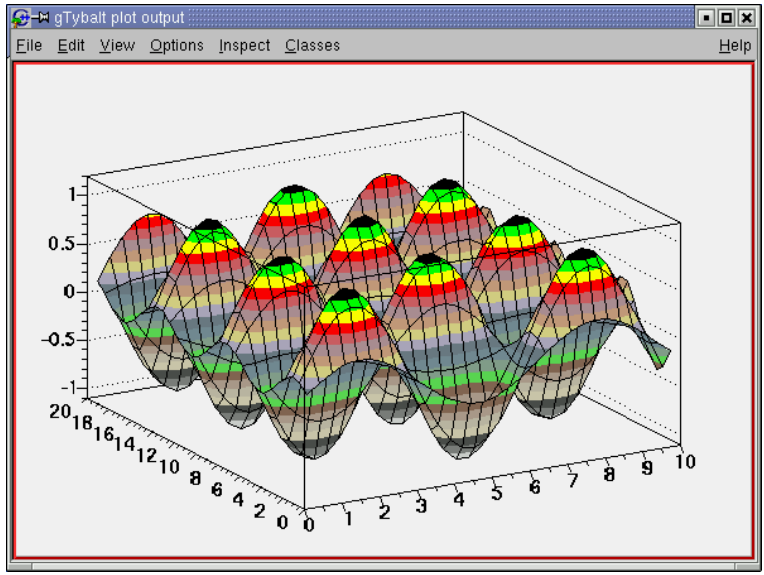


Fig. 4: The plot for the function $\sin(x)\sin(y)$ *for x from* 0 *to* 10 *and y from* 0 *to* 20.

2.4 Numerical integration

Functions can be integrated numerically by Monte Carlo integration. For example to evaluate the integral

$$\int_0^1 dx \int_0^1 dy \int_0^1 dz\, xyz \tag{1}$$

one types

```
gTybalt> symbol x("x"), y("y"), z("z");
gTybalt> ex f = x*y*z;
gTybalt> ex g = intnum(f,lst(x,y,z),lst(0,0,0), lst(1,1,1));
gTybalt> print(g);
0.12500320720479463077
```

The result of the integration can also be accessed with the help of the global variable `gTybalt_int_res`. In addition the global variables `gTybalt_int_err` and `gTybalt_int_chi2` give information on the error and the χ^2. For our example, one gets

```
gTybalt> print(gTybalt_int_res);
0.125003
```

```
gTybalt> print(gTybalt_int_err);
5.26234e-06
gTybalt> print(gTybalt_int_chi2);
0.385639
```

The Monte Carlo integration uses the adaptive algorithm VEGAS [11]. gTybalt uses the implementation from the GNU Scientific Library.

3 Nested sums

Let me start with a motivation from particle physics: Loop integrals occur in higher orders in perturbation theory. An example for a loop integral is given by the following three-point function

$$\begin{aligned} I &= \frac{\Gamma(1-2\varepsilon)}{\Gamma(1+\varepsilon)\Gamma(1-\varepsilon)^2} \left(-s_{123}\right)^{\nu_{123}-m+\varepsilon} \\ &\quad \int \frac{d^D k_1}{i\pi^{D/2}} \frac{1}{(-k_1^2)^{\nu_1}} \frac{1}{(-k_2^2)^{\nu_2}} \frac{1}{(-k_3^2)^{\nu_3}}, \end{aligned} \tag{2}$$

where $k_1 = k_2 + p_1 + p_2$, $k_2 = k_3 + p_3$ are the momenta of the particles. The external momenta p_j satisfy $p_1^2 = p_2^2 = p_3^2 = 0$, which says that the external particles are massless particles travelling with the speed of light. D is the dimension of space-time. It is convenient to take it as a a complex number, which lies in the neighbourhood of an integer number. The dimension is therefore parameterized as $D = 2m - 2\varepsilon$, where m is an integer and ε a small parameter. The other quantities are defined by $s_{123} = (p_1 + p_2 + p_3)^2$ and $\nu_{123} = \nu_1 + \nu_2 + \nu_3$. The exact meaning of all these quantities is not too important here. For the purpose here it is sufficient to state that we are interested in the value of this integral as an expansion in ε around the point $D = 2m$. A short calculation shows that this integral evaluates to the following hypergeometric function

$$\begin{aligned} I &= \frac{\Gamma(1-2\varepsilon)}{\Gamma(1+\varepsilon)\Gamma(1-\varepsilon)^2} \frac{1}{\Gamma(\nu_1)\Gamma(\nu_2)} \frac{\Gamma(m-\varepsilon-\nu_1)\Gamma(m-\varepsilon-\nu_{23})}{\Gamma(2m-2\varepsilon-\nu_{123})} \\ &\quad \times \sum_{n=0}^{\infty} \frac{\Gamma(n+\nu_2)\Gamma(n-m+\varepsilon+\nu_{123})}{\Gamma(n+1)\Gamma(n+\nu_{23})} (1-x)^n, \end{aligned} \tag{3}$$

where $x = s_{12}/s_{123}$. We are interested in the Laurent expansion in ε for given integer values of ν_1, ν_2, ν_3 and m. More complicated loop integrals lead to additional classes of infinite sums. Typical examples are the generalized hypergeometric functions

$${}_{J+1}F_J(a_1,...,a_{J+1};b_1,...,b_J;x) = \sum_{n=0}^{\infty} \frac{(a_1)_n ... (a_{J+1})_n}{(b_1)_n ... (b_J)_n} \frac{x^n}{n!}, \tag{4}$$

the first and second Appell functions defined by

$$F_1(a,b_1,b_2;c;x_1,x_2) = \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} \frac{(a)_{m_1+m_2}(b_1)_{m_1}(b_2)_{m_2}}{(c)_{m_1+m_2}} \frac{x_1^{m_1}}{m_1!}\frac{x_2^{m_2}}{m_2!},$$

$$F_2(a,b_1,b_2;c_1,c_2;x_1,x_2) = \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} \frac{(a)_{m_1+m_2}(b_1)_{m_1}(b_2)_{m_2}}{(c_1)_{m_1}(c_2)_{m_2}} \frac{x_1^{m_1}}{m_1!}\frac{x_2^{m_2}}{m_2!}, \tag{5}$$

or the Kampé de Fériet function defined by

$$S_1(a_1,a_2,b_1;c,c_1;x_1,x_2) = \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} \frac{(a_1)_{m_1+m_2}(a_2)_{m_1+m_2}(b_1)_{m_1}}{(c)_{m_1+m_2}(c_1)_{m_1}} \frac{x_1^{m_1}}{m_1!}\frac{x_2^{m_2}}{m_2!}. \tag{6}$$

$(a)_n = \Gamma(n+a)/\Gamma(a)$ denotes a Pochhammer symbol. In all these cases one seeks the Laurent expansion around integer values around the parameters which occur in the Pochhammer symbols. The calculation of loop integrals is equivalent to the task of performing the Laurent expansion. Note that computing the Laurent expansion is a purely mathematical problem, which can be formulated without any reference to particle physics. We developed systematic algorithms to perform this task. The key point is the introduction of mathematical objects, which have a particular form of nested sums and are called Z-sums [12–14]:

$$Z(n;m_1,...,m_k;x_1,...,x_k) = \sum_{n\geq i_1>i_2>...>i_k>0} \frac{x_1^{i_1}}{{i_1}^{m_1}} \cdots \frac{x_k^{i_k}}{{i_k}^{m_k}} \tag{7}$$

The Z-sums contain as subsets some other classes of special functions. If the sums go to Infinity ($n=\infty$) the Z-sums are the multiple polylogarithms:

$$Z(\infty;m_1,...,m_k;x_1,...,x_k) = \mathrm{Li}_{m_1,...,m_k}(x_1,...,x_k). \tag{8}$$

For $x_1 = ... = x_k = 1$ the Z-sums reduce to the Euler-Zagier sums:

$$Z(n;m_1,...,m_k;1,...,1) = Z_{m_1,...,m_k}(n). \tag{9}$$

For $n=\infty$ and $x_1 = ... = x_k = 1$ the sum is a multiple ζ-value:

$$Z(\infty;m_1,...,m_k;1,...,1) = \zeta(m_1,...,m_k). \tag{10}$$

The Z-sums form a Hopf algebra. In particular, any product of two Z-sums with the same upper summation bound n can always be reduced to a sum of single Z-sums. For example:

$$Z_{11}(n)\cdot Z_1(n) = Z_{21}(n) + Z_{12}(n) + 3\,Z_{111}(n). \tag{11}$$

The multiplication property is at the core of all our algorithms. Euler-Zagier sums occur in the expansion of Gamma functions:

$$\begin{aligned}\Gamma(n+\varepsilon) &= \Gamma(1+\varepsilon)\Gamma(n) \\ &\times \left(1+\varepsilon Z_1(n-1)+\varepsilon^2 Z_{11}(n-1)+\varepsilon^3 Z_{111}(n-1)+\right. \\ &\left. ...+\varepsilon^{n-1} Z_{11...1}(n-1)\right).\end{aligned} \tag{12}$$

The usefulness of the Z-sums lies in the fact, that they interpolate between multiple polylogarithms and Euler-Zagier sums. In addition, the interpolation is compatible with the algebra structure. Apart from the multiplication algorithm we have three additional operations on Z-sums if we allow expressions of the form

$$\frac{x_0^n}{n^{m_0}} Z(n;m_1,...,m_k;x_1,...,x_k). \tag{13}$$

Then the following convolution product

$$\sum_{i=1}^{n-1} \frac{x^i}{i^m} Z(i-1;...) \frac{y^{n-i}}{(n-i)^{m'}} Z(n-i-1;...) \tag{14}$$

can again be expressed in terms of expressions of the form (13). In addition there is a conjugation, e.g. sums of the form

$$-\sum_{i=1}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) (-1)^i \frac{x^i}{i^m} Z(i;...) \tag{15}$$

can also be reduced to terms of the form (13). The name conjugation stems from the following fact: To any function $f(n)$ of an integer variable n one can define a conjugated function $C \circ f(n)$ as the following sum

$$C \circ f(n) \quad = \quad \sum_{i=1}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) (-1)^i f(i). \tag{16}$$

Then conjugation satisfies the following two properties:

$$\begin{aligned} C \circ 1 &= 1, \\ C \circ C \circ f(n) &= f(n). \end{aligned} \tag{17}$$

Finally there is the combination of conjugation and convolution, e.g. sums of the form

$$-\sum_{i=1}^{n-1} \left(\begin{array}{c} n \\ i \end{array} \right) (-1)^i \frac{x^i}{i^m} Z(i;...) \frac{y^{n-i}}{(n-i)^{m'}} Z(n-i;...) \tag{18}$$

can also be reduced to terms of the form (13). These four algorithms are sufficient to obtain the Laurent expansion of the functions in eq. (4) to (6). They are implemented in the library "nestedsums". When gTybalt is compiled with the nestedsums library, gTybalt provides an interface to expand these transcendental functions in a small parameter. For example, a hypergeometric function can be expanded as follows:

```
gTybalt> symbol x("x"), eps("epsilon");
gTybalt> transcendental_fct_type_A
         F21(x,lst(1,-eps),lst(1-eps),lst(1-eps),
         lst(1,-eps));
gTybalt> ex f = F21.set_expansion(eps,5);
gTybalt> rawprint(f);
-Li(3,x)*epsilon^3-Li(2,x)*epsilon^2-Li(4,x)
*epsilon^4-Li(1,x)*epsilon+Z(Infinity)
```

This expands the hypergeometric function ${}_2F_1(1,-\varepsilon;1-\varepsilon;x)$ in ε up to order 5 and agrees with the known expansion

$$
\begin{aligned}
&{}_2F_1(1,-\varepsilon;1-\varepsilon;x) = \\
&\quad 1-\varepsilon\,\mathrm{Li}_1(x)-\varepsilon^2\,\mathrm{Li}_2(x)-\varepsilon^3\,\mathrm{Li}_3(x)-\varepsilon^4\,\mathrm{Li}_4(x)+O(\varepsilon^5).
\end{aligned}
\tag{19}
$$

Z(Infinity) represents the unit element in the algebra of Z-sums and is equal to 1.

4 Applications

In this chapter I would like to discuss a few applications. First of all, the two programs were developed for the calculation of quantum loop corrections in elementary particle physics. A specific process is the process $e^+e^- \to 3$ jets. In a very simple picture this process involves an incoming electron and an incoming positron, which annihilate into a photon or Z-boson. The photon or Z-boson subsequently splits into a quark-antiquark-pair. Then either the quark or the antiquark radiates off an additional gluon. These three particles (the quark, the antiquark and the gluon) are seen in the detector as three hadronic jets. This process has been measured at the LEP-experiment at CERN in Geneva. From this process one can extract the value of the strong coupling constant. The strong coupling constant is a fundamental parameter of nature. We know that in nature there are four fundamental forces: the gravitational, the electromagnetic, the weak and the strong force. The strength of the strong force is described by the strong coupling constant. To extract the numerical value from experiment one needs a precise calculation for this process. At the per cent level the computation of the two-loop scattering amplitude is re-

quired. This was a challenging task and tackled with the programs presented above [15].

As a second application these computations have triggered increased interest into multiple polylogarithms. These are the functions which appear in the results of a calculation. At the end of the day of an analytic calculation physicists would like to get a number. This requires a method for the numerical evaluation of multiple polylogarithms. Multiple polylogarithms are functions of several complex variables with a rather complicated branch cut structure. We developed routines for the numerical evaluation of these functions for all values of the argument [16].

A third application goes into the direction of number theory: In massless quantum field theories there is one two-loop two-point function, from which all other two-loop two-point functions can be derived. The dependence on the momentum squared of this integral can trivially be factored out, so the remainder is a pure number. For a long time it has been an open question what type of numbers occur in this result. To cite a paper from the year 2002 [17]: *"It is one of the many scandals of our limited understanding of the analytical content of perturbative quantum field theory that, despite many years of intense effort, we still do not know whether multiple zeta values suffice for even the Taylor expansion of the two–loop integral."* Using the mathematical structure of our algorithms, we could prove that to all orders in ε multiple zeta values are sufficient [18].

As a further application I would like mention recent developments in mathematics: Multiple polylogarithms and multiple zeta values figure prominently in the theory of mixed Tate motives. The connection between physics and mathematics has been worked out in a recent paper by Bloch, Kreimer and Esnault [19].

Finally, it should be noted that commercial computer algebra systems are not able to expand hypergeometric functions, if the expansion parameter occurs in the Pochhammer symbols. They start now to implement our algorithms [20].

A Design of the program "gTybalt"

This section gives some technical details on the design of the program "gTybalt" and serves as a guide to the source code. After a general overview of the system I discuss two technical points concerning threads and dynamic loading, where a few explanations might be useful to understand the source code.

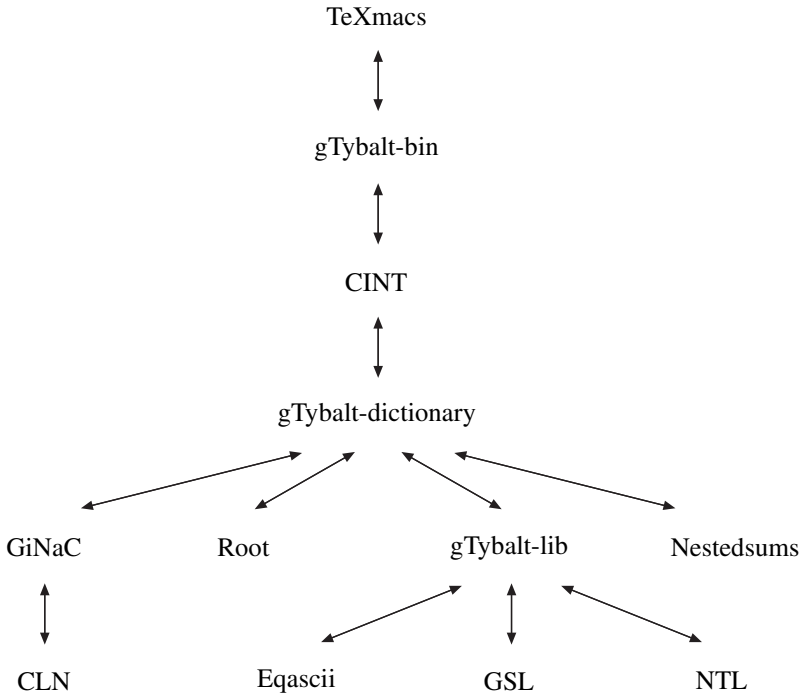


Fig. 5: Structural overview for gTybalt.

A.1 Structural overview

A structural overview for gTybalt is shown in fig. 5. gTybalt consists of three parts, labelled `gTybalt-bin`, `gTybalt-dictionary` and `gTybalt-lib`, which ensure communication between the different modules on which gTybalt is based. The first part, `gTybalt-bin` is either called from TeXmacs (in TeXmacs-mode) or directly from the shell (in text mode) and implements an event loop. This program reads input from the keyboard, sends the commands to the C++ interpreter CINT for execution and directs the output either to TeXmacs or to a text window.
The program CINT interprets the commands. For this purpose it uses a library called
`gTybalt-dictionary`, which can be thought of as a look-up table where to find the actual implementations of the encountered function calls. The source code for this library is generated automatically during the build phase of gTybalt. A file "`LinkDef.h`" specifies which functions and classes are included into this library. The library is then generated from the header files for these

functions and classes. The CINT interpreter is not 100% standard C++ compatible and there are certain constructs, which cannot be processed by CINT. Therefore the header files for the GiNaC-library are first copied to a temporary directory and then processed by a perl script, which comments out any parts which cannot be fed into CINT.
Finally, the library `gTybalt-lib` is an ordinary library, defining gTybalt-specific functions like `print`, `factorpoly` or `intnum`. It depends in turn on other libraries, like Eqascii, GSL or NTL, which are however not visible in the interactive interface.

A.2 Threads and the plotting routine

When plotting a function, it is desirable to have the window with the plot appearing on the screen, but at the same time still be able to work in the main window of gTybalt. Since there are now two possible actions which the user can take (e.g. typing new commands in the main window and modifying the plot inside the window with the plot) this is implemented using different threads. When starting gTybalt, the program will create a separate thread, which waits on a condition that a function should be plotted. When a plotting command is issued, CINT invokes a function, which just prepares some variables for the plot, signals that there is something to be plotted and then returns. Therefore after the return of this function the user can issue new commands in the main window of gTybalt. The thread waiting on the condition for plotting a function will wake up, plot the function and provide an event handler for events concerning the window with the plot. Therefore the user can now take actions in both the main window for gTybalt and the window with the plot. Once the window with the plot is cleared (by choosing from the menu-bar of the plot "File → Quit ROOT") the thread for plots will fall into sleep again and wait till another plotting command is issued. Thread safety is guaranteed by copying the relevant expressions for the function to be plotted to global variables and by the reference counting mechanism of GiNaC: The expression to be plotted will be pointed at by at least one (global) variable, therefore it will not be modified. While a plot is displayed on the screen, any command to plot another function will be ignored. The user must first clear the window with the plot.

A.3 Dynamic loading of modules and numerical integration

The default behaviour for numerical evaluation of a function uses the arbitrary precision arithmetic provided by the CLN library. For Monte Carlo integration, where a function needs to be evaluated many times, this is quite slow and therefore inefficient. It is also not needed, since statistical errors and not

rounding errors tend to dominate the error of the final result. Therefore a different approach has been implemented for the numerical Monte Carlo integration: The function to be integrated is first written as C code to a file, this file is then compiled with a standard C compiler and the resulting executable is loaded dynamically (e.g. as a "plug-in") into the memory space of gTybalt and the Monte Carlo integration routine uses this compiled C function for the evaluations.

References

[1] S. Weinzierl, Comput. Phys. Commun. **145**, 357 (2002), math-ph/0201011.
[2] S. Weinzierl, Comput. Phys. Commun. **156**, 180 (2004), cs.sc/0304043.
[3] J. van der Hoeven, Cahiers GUTenberg **39-40**, 39 (2001);
J. van der Hoeven, "TeXmacs" (1999), http://www.texmacs.org.
[4] P. Borys, "eqascii" (2001), http://dione.ids.pl/~pborys/software/linux.
[5] M. Goto, "C++ Interpreter - CINT", CQ publishing, ISBN 4-789-3085-3 (in japanese);
M. Goto, "CINT", http://root.cern.ch/root/Cint.html.
[6] C. Bauer, A. Frink, and R. Kreckel, J. Symbolic Computation **33**, 1 (2002), cs.sc/0004015;
"GiNaC library", http://www.ginac.de.
[7] B. Haible, "CLN library" (1999), http://www.ginac.de/CLN.
[8] R. Brun and F. Rademakers, Nucl. Inst. & Meth. in Phys. Res. **A389**, 81 (1997);
"Root", http://root.cern.ch.
[9] M. Galassi et al., "GNU scientific library", http://sources.redhat.com/gsl.
[10] V. Shoup, "NTL library" (1990), http://www.shoup.net.
[11] G. P. Lepage, J. Comput. Phys. **27**, 192 (1978).
[12] S. Moch, P. Uwer, and S. Weinzierl, J. Math. Phys. **43**, 3363 (2002), hep-ph/0110083.
[13] S. Weinzierl, (2003), hep-th/0305260.
[14] S. Weinzierl, J. Math. Phys. **45**, 2656 (2004), hep-ph/0402131.
[15] S. Moch, P. Uwer, and S. Weinzierl, Phys. Rev. **D66**, 114001 (2002), hep-ph/0207043.
[16] J. Vollinga and S. Weinzierl, Comput. Phys. Commun. **167**, 177 (2005), hep-ph/0410259.
[17] D. J. Broadhurst, Nucl. Phys. Proc. Suppl. **116**, 432 (2003), hep-ph/0211194.
[18] I. Bierenbaum and S. Weinzierl, Eur. Phys. J. **C32**, 67 (2003), hep-ph/0308311.
[19] S. Bloch, H. Esnault, and D. Kreimer, (2005), math.AG/0510011.
[20] T. Huber and D. Maitre, Comput. Phys. Commun. **175**, 122 (2006), hep-ph/0507094.

Anschriften der Autoren

Dipl.-Ing. Claas Cornelius
Institut für Angewandte Mikroelektronik und Datentechnik
Universität Rostock, Haus 1
Richard-Wagner Str. 31, 18119 Rostock-Warnemünde
E-Mail: claas.cornelius@uni-rostock.de

Dr. Frank Grassert
Institut für Angewandte Mikroelektronik und Datentechnik
Universität Rostock, Haus 1
Richard-Wagner Str. 31, 18119 Rostock-Warnemünde
E-Mail: frank.grassert@uni-rostock.de

Dr. Michael Habeck
Max-Planck-Institut fuer Entwicklungsbiologie und Max-Planck-Institut fuer Biologische Kybernetik, Tübingen
Spemannstrasse 35, 72076 Tuebingen
E-Mail: michael.habeck@tuebingen.mpg.de

Dr. Rafał Mantiuk
Max-Planck-Institut für Informatik
Abteilung Computer Grafik
Stuhlsatzenhausweg 85, 66123 Saarbrücken
E-Mail: mantiuk@mpi-inf.mpg.de

Dr. Sven Meyer zu Eissen
Bauhaus-Universität Weimar
Fakultät Medien, Mediensysteme
Bauhausstraße 11, 99423 Weimar
E-Mail: sven.meyer-zu-eissen@medien.uni-weimar.de

Dr. Wolfgang Rieping
Department of Biochemistry, University of Cambridge
80 Tennis Court Road, Cambridge CB2 1GA, UK
E-Mail: wolfgang.rieping@bioc.cam.ac.uk

Dipl.-Ing. Frank Sill
Institut für Angewandte Mikroelektronik und Datentechnik
Universität Rostock, Haus 1
Richard-Wagner Str. 31, 18119 Rostock-Warnemünde
E-Mail: frank.sill@uni-rostock.de

Prof. Dr. Benno Stein
Bauhaus-Universität Weimar
Fakultät Medien, Mediensysteme
Bauhausstraße 11, 99423 Weimar
E-Mail: benno.stein@medien.uni-weimar.de

Dr Johannes Söding
Max-Planck-Institut für Entwicklungsbiologie,
Spemannstrasse 35, 72076 Tübingen
E-Mail : johannes.soeding@tuebingen.mpg.de

Prof. Dr. Dirk Timmermann
Institut für Angewandte Mikroelektronik und Datentechnik
Universität Rostock, Haus 1
Richard-Wagner Str. 31, 18119 Rostock-Warnemünde
E-Mail: dirk.timmermann@uni-rostock.de

Prof. Dr. Stefan Weinzierl
Institut für Physik (ThEP),
Universität Mainz
Staudinger Weg 7, 55099 Mainz
E-Mail: stefanw@thep.physik.uni-mainz.de

In der Reihe GWDG-Berichte sind zuletzt erschienen:

Nr. 40 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1994
1995

Nr. 41 *Brinkmeier, Fritz* (Hrsg.):
Rechner, Netze, Spezialisten. Vom Maschinenzentrum zum Kompetenzzentrum – Vorträge des Kolloquiums zum 25jährigen Bestehen der GWDG
1996

Nr. 42 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1995
1996

Nr. 43 *Wall, Dieter* (Hrsg.):
Kostenrechnung im wissenschaftlichen Rechenzentrum – Das Göttinger Modell
1996

Nr. 44 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1996
1997

Nr. 45 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
13. DV-Treffen der Max-Planck-Institute – 21.-22. November 1996 in Göttingen
1997

Nr. 46 **Jahresberichte 1994 bis 1996**
1997

Nr. 47 *Heuer, Konrad, Eberhard Mönkeberg und Ulrich Schwardmann:*
Server-Betrieb mit Standard-PC-Hardware unter freien UNIX-Betriebssystemen
1998

Nr. 48 *Haan, Oswald* (Hrsg.):
Göttinger Informatik Kolloquium – Vorträge aus den Jahren 1996/97
1998

Nr. 49 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
IT-Infrastruktur im wissenschaftlichen Umfeld – 14. DV-Treffen der Max-Planck-Institute, 20.-21. November 1997 in Göttingen
1998

Nr. 50 *Gerling, Rainer W.* (Hrsg.):
Datenschutz und neue Medien – Datenschutzschulung am 25./26. Mai 1998
1998

Nr. 51 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1997
1998

Nr. 52 *Heinzel, Stefan und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1998
1999

Nr. 53 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Internet- und Intranet-Technologien in der wissenschaftlichen Datenverarbeitung – 15. DV-Treffen der Max-Planck-Institute, 18. - 20. November 1998 in Göttingen
1999

Nr. 54 *Plesser, Theo und Helmut Hayd* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1999
2000

Nr. 55 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Neue Technologien zur Nutzung von Netzdiensten – 16. DV-Treffen der Max-Planck-Institute, 17. - 19. November 1999 in Göttingen
2000

Nr. 56 *Plesser, Theo und Helmut Hayd* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2000
2001

Nr. 57 *Hayd, Helmut und Rainer Kleinrensing (Hrsg.)*
17. und 18. DV-Treffen der Max-Planck-Institute,
22. - 24. November 2000, 21. – 23. November 2001 in Göttingen
2002

Nr. 58 Macho*, Volker und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2001
2003

Nr. 59 *Suchodoletz, Dirk von*:
Effizienter Betrieb großer Rechnerpools – Implementierung am Beispiel des Studierendennetzes an der Universität Göttingen
2003

Nr. 60 *Haan, Oswald (Hrsg.)*:
Erfahrungen mit den IBM-Parallelrechnersystemen RS/6000 SP und pSeries690
2003

Nr. 61 *Rieger, Sebastian*:
Streaming-Media und Multicasting in drahtlosen Netzwerken – Untersuchung von Realisierungs- und Anwendungsmöglichkeiten
2003

Nr. 62 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2002
2003

Nr. 63 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2003
2004

Nr. 64 *Koke, Hartmut* (Hrsg.):
GÖ* - Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ*-Vorantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation“
2004

Nr. 65 *Koke, Hartmut* (Hrsg.):
GÖ* - Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ*-Hauptantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation“
2004

Nr. 66 *Bussmann, Dietmar und Andreas Oberreuter* (Hrsg.):
19. und 20. DV-Treffen der Max-Planck-Institute
20.-22. November 2002
19.-21. November 2003 in Göttingen
2004

Nr. 67 *Gartmann, Christoph und Jochen Jähnke* (Hrsg.):
21. DV-Treffen der Max-Planck-Institute
17.-19. November 2004 in Göttingen
2005

Nr. 68 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2004
2004

Nr. 69 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2005
2005

Nr. 70 *Gartmann, Christoph und Jochen Jähnke* (Hrsg.):
22. DV-Treffen der Max-Planck-Institute
16.-18. November 2005 in Göttingen
2006

Nr. 71 *Hermann, Klaus und Jörg Kantel* (Hrsg.):
23. DV-Treffen der Max-Planck-Institute
15.-17. November 2006 in Berlin
2007

Nr. 72 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2006
2007

Nähere Informationen finden Sie im Internet unter
`http://www.gwdg.de/forschung/publikationen/gwdg-berichte/index.html`