Dissemination and Visualisation of Biological Data

BERNAT GEL MORENO

PhD Thesis supervised by

Xavier Messeguer Peypoch

July 2014

Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics

Contents

Acknowledgements

Aquesta tesis no hauria estat possible sense l'ajuda i el suport de molta gent, i a tots ells m'agradaria donar-los les gràcies.

A tota la gent del laboratory d'oncologia molecular al clínic que em van obrir les portes a un camp que desconeixia totalment, començant per en Marià Monzó, que em va donar la oportunitat de començar a treballar com a bioinformàtic, a tots els companys del laboratori, la Sònia, la Rosa, l'Anna, la Sílvia, la Rut, en Gerardo, la Marina, la Carme, l'Aina i molt especialment a l'Alfons Navarro, per tot el que hem après junts, per les magnífiques hores de discussions i per ensenyar-me què és la passió per la recerca.

A tots els companys de l'IMPPC, per acollir-me i aguantar-me, la Imma, la Eli, en Josep, l'Ernest, en Carles, la Meri i evidentment la Raquel, l'Anna, la Yaiza, en Roberto i tota la resta de gent que fa que poguem estar treballant entre amics. I gràcies també a l'Edu Serra, per tot el que m'ha ensenyat i sobretot per la paciència que ha tingut amb aquesta tesis.

A totes les persones que m'han hagut d'aguantar al llarg d'aquests anys i que m'han convençut una vegada i una altra de seguir endavant, la Ciara, l'Anna, els dos Joans, la Júlia, la Maria i la Raquel i en Carlos. Als meus pares i a la meva germana, que han aguantat estoicament hores i hores de xerrera sobre temes d'interés més que dubtós i que han estat aquí sempre. I a la Sonia, que ha compartit amb mi aquest sprint final, i que és en gran mesura responsable que això finalment hagi arribat a bon port.

I evidentment, al meu director, en Xavier Messeguer, per la seva ajuda i guia.

A tots vosaltres, moltes gràcies.

Chapter 1

Introduction

With the recent advent of various waves of technological advances, the amount of biological data being generated has exploded. As a consequence of this data deluge, new challenges have emerged in the field of biological data management. In order to maximize the knowledge extracted from the huge amount of biological data produced it is of great importance for the research community that data dissemination, integration and visualisation challenges are tackled. Opening and sharing our data and working collaboratively will benefit the scientific community as a whole and to move towards that end, new developments, tools and techniques are needed.

1.1 Background

Technological advances and the data deluge

Recent technological advances have had a great impact on the amount of biological data generated. During the first years of the last decade, due to the generalization of high throughput microarray technologies -mainly gene expression microarrays but also lots of related technologies such as aCGH, ChIP-on-chip, methylation arrays-, a first data revolution happened. All of a sudden, it was possible for many labs to generate amounts of data never seen before on biology. While with older technologies it was possible to experimentally determine expression values for up to a few hundreds of genes and the process was time consuming and involved many manual operations, a single microarray chip would produce expression data for tens of thousands of genes in an almost fully automated process. It was possible, for the first time, to study the complete transcription profile of a cell or to get a broad view of the methylation status of the genome. In order to achieve such higher data throughputs, though, trade-offs had to be made on data quality and reliability, since arrays had higher noise levels and lower accuracy than low throughput techniques such as real-time PCR. Fortunately, statisticians and bioinformaticians developed a number of new normalization and analysis techniques to deal with such a noisy data. Many new and highly relevant results were produced using array-based technologies and most of them are still being used today for a number of experiments with great success.

Over the end of the last decade, however, while microarray use spread over the research world and was made available to most of the research groups, a second wave of even higher throughput technologies emerged. Bead arrays, where probes were not spotted to specific places but fixed to micrometer sized silica bead and randomly distributed over the array surface[1], delivered a higher throughput than traditional microarrays, but the real breakthrough came from what's known as Next Generation Sequencing (NGS) or massively parallel sequencing, a set of technologies giving the researchers the possibility to sequence huge amounts of DNA at a very low price per base.

Next Generation Sequencing is profoundly reshaping the landscape of available experiments, being now feasible to perform experiments at a scale and level of detail that would have been unfeasible just a few years ago. While the first human genome to be produced required a big international consortium, involved many different laboratories working at full capacity for several years and had a total cost of about $3 billions[2], with the machines found today in many genomic facilities it is possible to re-sequence a whole human genome for a few thousand dollars. The affordability of these systems has favored projects such as *The 1000 Genomes Project*[3]-an international collaboration to produce an extensive public catalog of human genetic variation-, and *The Cancer Genome Atlas*[4] (http://cancergenome.nih.gov/) -a collaborative project to study genetic variations and genomic alterations related to cancer-.

A new system presented on January 2012 by Life Technologies, the Ion Proton, will be, according to the Life Technologies, the company developing it, the first system to achieve the milestone of the $1000 genome. Being able to sequence a whole human genome in a few hours for as little as $1000 will open a new range of applications and will increase even more the pace of genomic data generation.

Newer technologies being developed and based on completely different sequencing procedures, such as Nanopore and other single molecule sequencing approaches, could be available in a few years which would mean a third large jump in the pace at which biological data is generated and heavily contribute to the worsening of the current data deluge.

Data repositories

In order to ensure long term storage for all the data generated with high throughput technologies and to make it available to the research community, public repositories were created. Two of the most widely used public repositories are Gene Expression Omnibus[5] from the NCBI and Array Express[6] developed at the EBI. In part thanks to the requirement of having deposited the raw data into one of those repositories in order to publish in many high profile journals, data from hundreds of thousands of experiments is freely available to anyone wanting to work with it. The availability of such amount of data has spurred the development of new analysis methods for high throughput data and made possible a number of meta-analysis.

Cooperation is key

Nowadays, many small research groups capable of producing important and interesting datasets lack the capabilities to analyze them fully. The release of those datasets, instead of keeping them closed on the laboratory to be used never again, can greatly increase their scientific value, since specilized analysis groups can apply deeper analysis techniques to extract further information from them, producing new results. In addition, as demonstrated by many meta-analysis performed using publicly available datasets, thanks to the availability of large amounts of public data it is possible to extract new information from its integrated analysis. Development of new algorithms, specially those with their roots on artificiall intelligence, greatly benefit from the availability of a big corpus of annotated datasets for training and testing purposes, giving new and better algorithms to biomedical sciences in return. None of these would be feasible without large amounts of biological data made freely and publicly available by a number of researchers around the world.

Publishing raw or minimally processed data along with the necessary metadata is necessary to ensure that complete reanalysis are possible. It also encourages correctness and reproducibility checks on that data by other researchers around the world, which is fundamental to assure the validity of published results. These checks and reanalysis would not be possible if no raw data was made available, since decisions taken very early in the data analysis process such as normalization may have a huge impact on the final results. Thus, releasing raw datasets for whole experiments and not only the final selected results have a beneficial impact on the whole research process. In addition, data can be much more valuable when taken into consideration along with other data of diverse origin, either of the same type or complementary data. Therefore, it is very important to be able to consistently integrate all this data.

Metadata and Data Integration

Integrated analysis of different data types is key to understand complex systems such as biological entities. Often, these different data types come from different sources and are produced by different means at different geographical locations. For example, one could match the position of genes in the genome as determined by Ensembl programmatically and cross this data with the position of SNPs determined by the aggregation of thousands of submissions to dbSNP at the NCBI and then use a web-service to check in a motif database whether a subset of the genes with SNPs at the promoter region have any specific motif and then study the correlation of these motifs with expression data from a GEO dataset. In doing all this multistep analysis you are taking advantage of the availability of all different kinds of public data but at the same time you are doing some important assumptions. For example, the reference genome assembly used to compute gene positions by Ensembl must be exactly the same than the one used by dbSNP and also the same used in the motif database. In addition, the cellular types and conditions used for the experiment from which the expression data was obtained should be relevant and adequate to test your biological hypothesis.

A key aspect to take into account when releasing data is metadata. Metadata should contain all information needed to fully understand the released data. Once data is fully understood it will be possible to decide whether it is suitable for its intended use and if any changes or adaptations are needed before using it along data from other sources. Thorough and accurate metadata is key to ensure that data is used correctly. In the process describe above, for example, describing genome over which the genes have been computed as Homo sapiens is clearly not enough and the exact genomic assembly should be stated, since there are some very notable differences between genomic alignments and even a minor feature misplacement of a few nucleotides might have a big impact in the final results.

Visualisation and Knowledge Discovery

A process at which humans excel is at detecting patterns. Pattern recognition is one of the great abilities of our brains and suitable representations of data may make these patterns apparent and help us identify them[7,8]. For example, if instead of a long list of aligned short reads or a even a list of regions with high read concentration as a result from a ChIP-seq experiment the analysis software creates a drawing depicting the actual peak shapes and positions in a region of the genome, it will be much easier to identify, for example, odd-shaped peaks -pointing us to bad quality data or to specially interesting regions- or patterns in their relative positions with other genomic features such as genes or promoter regions. Good representations of data greatly improves our capacity to better understand it and so, to generate hypothesis about it that can be later tested by other means.

In order to help improving data understanding and intuition, though, it is important that the representation used to depict the data is suitable for the type of data being represented and specially for the knowledge that has to be extracted from it. It is important that the right patterns are emphasized while other distracting aspects are hidden and that the right level of complexity is achieved. An oversimplified representation of data will likely hide important features, however, an overly complex representation won't hide most of the distracting features and effectively rendering some of the patterns undetectable.

In addition to the right complexity level, data representation has to be tailored to the kind of information we want to extract from it. In the ChIP-seq example, the data crunching and the graphical representation of it should be completely different if we are studying the spatial relation between peaks and other genomic features than if we are interested in biases produces by the sequencing technology used.

One of the most broadly used visualisation paradigms for genomic data are genomic browsers. A genomic browser (such as Ensembl[9], the UCSC Genome Browser[10], or GBrowse[11]) is capable of displaying different sets of features positioned on relative to a sequence along to the sequence itself. It is possible to explore the sequence and the features by moving around and zooming in and out in order to see exactly the desired region. The interactivity of this process and so the ease given to the user to perform such an exploratory process varies from a browser to the other, but is quite limited in most of those based in web technologies.

DAS: The Distributed Annotation System

The Distributed Annotation System (DAS) is a protocol designed to publish and integrate annotations on biological entities, mostly sequences, in a distributed fashion. DAS is structured as a client-server system where the end user uses a client to retrieve data from one or more servers and merge, process and visualise it. DAS development was started in 2001 by WormBase[12] as a way to distribute and share its genomic annotations among its users and was soon adopted by the Ensembl project[13] and other databases. Currently a central registry, the DAS Registry[14], has more than 1600 data sources registered and these sources have been created by more than 50 institutions. Data available through DAS includes most of the data available at Ensembl and UCSC databases as well as hundreds of sources created by other independent groups.

The idea behind DAS is that many different laboratories and groups might have relevant but partial knowledge regarding a biological entity, such as a genomic sequence. To combine, visualise and explore all this data jointly, it should not be necessary to merge it all in a central database but it should be possible to merge it *on-the-fly* by the software being used by the end user. This approach has some interesting advantages such as being highly dynamical with respect to the set of data sources being used -i.e. the end user could be able to select any compatible source, including those created by himself, to be added to the view instead of having to ask to the hypothetical central database to add that data and make it available. With data being under control of those producing it, new releases, updates and corrections can happen more often, without being tied to the common release schedule of a central database. Such freedom, though, comes at a price, and some problems are attached to the distributed approach: the end user software is, at the end, the one deciding what data sources make sense to use together, which is the best way to represent the retrieved data and how should that

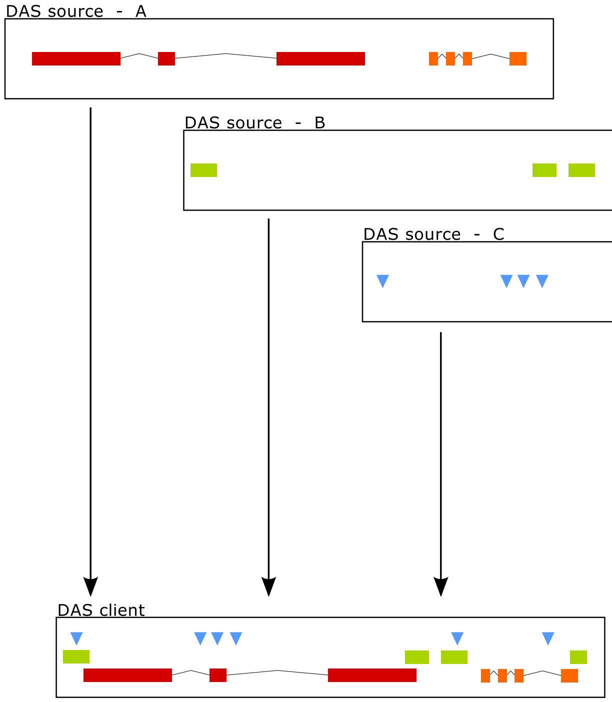


Figure 1.1 A schematic view of DAS. Different DAS servers offer a number of DAS sources (DAS source A, B and C) with different types of information annotating the same biological entity. A DAS client is able to access these data sources, retrieve data from them and create an integrated representation.

data exactly be merged. Different technological solutions have been adopted in DAS to overcome these problems, as will be briefly seen in this section.

A cartoon explaining the approach taken by DAS is shown in figure 1.1. For example, a research group, *Group A*, might have gene position information derived from various sources, another group, *Group B*, might have results from a ChIP-seq analysis of a transcription factor and finally a third group, *Group C* might have data related to the somatic mutations found in a type of cancer. If these three groups are publishing their data using DAS it is possible to integrate their data into one single integrated representation that could be more useful than any of the three sources of data by themselves. Additionally, the end user could set-up a fourth data source, *User Defined*, with gene expression data, and seamlessly integrate it with the data coming from the third party sources.

To ensure that it is possible to seamlessly integrate data from a number of different sources, at least three things are needed: a standard data protocol, a standard data format and a standard data semantics.

DAS is a client-server protocol, with data sources being the servers and the end-user software integrating the data being the client. It has been built in top of HTTP and takes advantage of HTTP headers -including custom headers- and HTTP status codes to manage the communication between clients and servers[15,16]. The DAS protocol is a stateless protocol that can be seen as an example of a Representational State Transfer (REST)[17] architecture. No state is saved between two consecutive requests to a server, all information resources available are uniquely identified by a structured URI and the responses are XML documents. One of the advantages of such a design is that DAS can work transparently with any network element such as proxies, caches and firewalls.

A DAS server, is an instance of a software accepting DAS requests at a specific URL. A single DAS server contains one or more DAS sources, which are the logical entries serving highly related data. For example, a group could setup a server for human data with three sources, one with information on genes, exons and UTR's, another one with data about SNPs and third with miRNA data. Servers and sources accept a few different commands, some optional and some compulsory for every server- instructing them to return different kinds of data and metadata - *sources* will retrieve a list of the sources in the server, *features* will return the actual annotations, etc.

To retrieve data from a server, a client issues a formatted URL request to the server and it returns an XML document. The URL format is well defined and contains different parts including the server URL, the source name, the command and any parameters it needs. Figure 1.2 shows an example of a DAS request: http://www.server.com/foo/ is the base URL of the DAS server, `sourcename` is the name of the source in the server, the command is `features` and a single parameter, `segment=1:1000,2000`, determines exactly which features are requested based on the specified region.

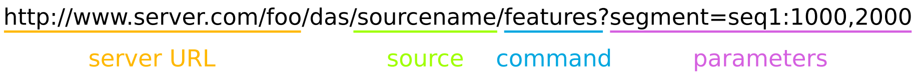
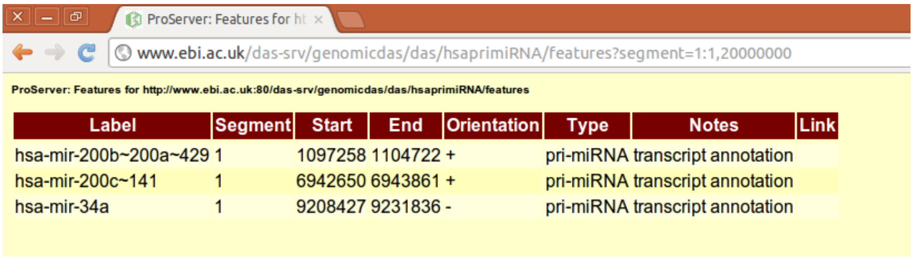

http://www.server.com/foo/das/sourcename/features?segment=seq1:1000,2000
server URL
source
command
parameters

Figure 1.2 An example of a DAS request. This DAS request has its different parts marked in different colors. This request to the source `sourcename` of the server `http://www.server.com/foo` will retrieve all annotation -`features`- overlapping the region starting at position 1000 to and ending at 2000 of the biological sequence identified by `seq1`.

The response to a successful DAS request is an XML document based on the formats defined by the DAS specification. There is a well defined XML format for each valid command and all servers are required to build a response strictly following these formats. The use of such standard formats ensures that all clients will be able to understand the responses from any valid DAS server and so that in order to include data from any additional server the only *a priori* information needed about the server is its URL. Combining this capability with the existence of a central DAS registry where clients can obtain a list of the available DAS servers and sources gives DAS clients a great flexibility in including new data to analysis


ProServer: Features for ht
www.ebi.ac.uk/das-srv/genomicdas/das/hsaprimiRNA/features?segment=1:1,20000000

ProServer: Features for http://www.ebi.ac.uk:80/das-srv/genomicdas/das/hsaprimiRNA/features

Label	Segment	Start	End	Orientation	Type	Notes	Link
hsa-mir-200b~200a~429	1	1097258	1104722	+	pri-miRNA	transcript annotation	
hsa-mir-200c~141	1	6942650	6943861	+	pri-miRNA	transcript annotation	
hsa-mir-34a	1	9208427	9231836	-	pri-miRNA	transcript annotation	

Figure 1.3 Example of a DAS response. The list of the miRNAs in the first 20Mb of human chromosome 1 returned by the EBI's DAS servers as rendered by a web browser with XSLT support.

and visualisations.

One additional feature of DAS is that since requests are simple URLs and responses are plain text documents it is possible to query DAS servers from a broad range of clients. If one would like to get a list of all the microRNAs in the first 20Mb of chromosome 1 in humans, firing a standard web browser and accessing the URL

http://www.ebi.ac.uk/das-srv/genomicdas/das/hsaprimiRNA/features?segment=1:1,20000000

would return a simple XML document with the list of the three primary microRNAs encoded in that region: *hsa-mir-200b*, *hsa-mir-200c* and *hsa-mir34a*. Since the XML document includes a link to an XSLT file, the browser is able to produce a nice human readable representation of the information (Figure 1.3) but the document returned is exactly what is depicted in Figure 1.4.

The REST style API allows easy and raw manual access to DAS servers using a web browser -useful for beginners as well as for testing and debugging purposes- or programmatically from any language with HTTP support or even from the command line using `wget` and the like, a useful option to develop simple data analysis scripts. All these methods, though, are lacking with respect to a full-blown DAS client able to retrieve and combine data from many different sources.

The third aspect of the DAS system is semantics. In addition to the exact syntax of the DAS XML documents, the DAS specification defines a minimal semantic framework so it's possible for clients to interpret them. The semantics defined by the specification are minimal and flexible enough to accommodate for different data types and categories, but without them it would be much more difficult to integrate data from different DAS servers. It is defined, for example, that DAS features are related to sequences, either to the whole sequence -e.g. annotating a protein sequence with a pubmed link- or to specific regions of the sequence -e.g. the pfam domains found in that same protein sequence-, that a feature defines at most a single region of the sequence -it is not possible for a feature to have more than one start and one end-, etc.

In order to be able to integrate data from various sources, it is very impor-

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="features.xsl"?>
<!DOCTYPE DASGFF SYSTEM
               "http://www.biodas.org/dtd/dasgff.dtd">
<DASGFF>
  <GFF version="1.01" href="http://www.ebi.ac.uk:80/das-
srv/genomicdas/das/hsaprimiRNA/features">
    <SEGMENT id="1" version="1.0" start="1" stop="20000000">
      <FEATURE id="hsa-mir-34a" label="hsa-mir-34a">
        <TYPE id="pri-miRNA">pri-miRNA</TYPE>
        <START>9208427</START>
        <END>9231836</END>
        <METHOD id="miRBaseGenomics">miRBaseGenomics</METHOD>
        <ORIENTATION>-</ORIENTATION>
        <NOTE>transcript annotation</NOTE>
      </FEATURE>
      <FEATURE id="hsa-mir-200c~141" label="hsa-mir-200c~141">
        <TYPE id="pri-miRNA">pri-miRNA</TYPE>
        <START>6942650</START><END>6943861</END>
        <METHOD id="miRBaseGenomics">miRBaseGenomics</METHOD>
        <ORIENTATION>+</ORIENTATION>
        <NOTE>transcript annotation</NOTE>
      </FEATURE>
      <FEATURE id="hsa-mir-200b~200a~429" label="hsa-mir-200b~200a~429">
        <TYPE id="pri-miRNA">pri-miRNA</TYPE>
        <START>1097258</START>
        <END>1104722</END>
        <METHOD id="miRBaseGenomics">miRBaseGenomics</METHOD>
        <ORIENTATION>+</ORIENTATION>
        <NOTE>transcript annotation</NOTE>
      </FEATURE>
    </SEGMENT>
  </GFF>
</DASGFF>
```

Figure 1.4 Raw XML DAS response. The XML document returned by the EBI's DAS servers when asked about the miRNAs in the first 20Mb of human chromosome 1.

tant to clearly define which are the sequences being annotated. In DAS this is achieved via the definition of coordinates systems. A coordinate system is a set of sequences related in some way. Sequence identifiers must be unique only in its coordinates system. For example, for genomic annotations one assembly of the set of chromosomes from a species is usually a coordinates system and for proteomic annotations, the coordinates system is the pool of all known protein sequences. In addition, a coordinates system establishes the sequence identifiers to be used, the entry points. Hence, once a source is associated to a coordinates system it is trivial to merge with data from other sources since they are all annotating exactly the same sequences. For example, the sequence 1 -the chromosome 1- from the assembly GRCh37 of *Homo sapiens* is unique and stable, and can not be confused with any other chromosomal sequence or P21359 in the coordinates system based on UniProt[18] uniquely identifies the human Neurofibromin aminoacid sequence.

Additional information on the DAS protocol, formats and semantics can be found in chapter 3 and in the DAS specification[16].

One of the mottos behind the design of DAS was *"Dumb server, Smart Client"*. The idea behind it is that it should be really easy for anyone with an interesting dataset to make it available via DAS and that as much as possible the complexities of the system should be moved to the client. There are a number of DAS clients available today, web based or stand-alone software and both general purpose and specialized. For example, the Ensembl genome browser, used daily by thousands of people, is a DAS client, which allows its users to add any data in a DAS server, even their own, to be merged into the Ensembl visualisation. Another widely used software, the Integrative Genomic Viewer (IGV)[19], is also able to display DAS tacks alongside local NGS data. In addition, since DAS client libraries are available in different programming languages, it is feasible to create customized client software answering specific needs a user or a group of users can have.

1.2 Motivation and Contributions

Distributing biological data in a way that makes it easy to use and integrate with other data sources is not as easy as it would be desirable and scientists have to face some challenges when doing it. Similarly, integration and visualisation of biological data could be an easier process giving more interactive results to encourage data exploration.

DAS can be a helpful tool to tackle some of these challenges and in this thesis we have produced different software tools and prototypes aimed at that.

1.2.1 Dissemination

Motivation

If a research group wanted to publish a dataset they had produced so anyone could access and use it, different options would be available for different kinds of data. For some data types, big repositories have been created and do a great job on storing and distributing them. GEO and Array Express, for example, are a great way of distributing data generated with expression microarrays and some other microarray based techniques. If the dataset is not of any of these few selected data types, however, distributing it in a useful way may become a much harder job.

Many biological data can be interpreted as annotations positioned on sequences, specially processed data and final results, but no easy and useful way of distributing such datasets is available. Making the text files available on FTP servers or provide them as supplementary material linked to a journal paper might not be enough to make them useful, since metadata is not guaranteed, file formats and semantics are not well established and no programmatic access is available, forcing researchers to manually download and manipulate them to make them mergeable with data from other sources.

The Distributed Annotation System may be a suitable solution for this, since it is specifically designed to distribute this kind of data, using standardized formats and semantics and ensuring sufficient metadata is provided so integration of multi-source data is possible. The problem with this approach is that setting up a DAS server imposes some requirements not met by many research groups. In addition to the obvious requirement of a server facing the internet, setting up a DAS server usually implies installing a database, a bit of *data mashing* to adapt it to DAS and load the database and even a bit of programming, since the data access layer connecting the DAS server to the database is usually left for the user to implement. While these requirements could be easily met by computational biology labs and the like, for purely wet labs they may become problems impossible to overcome and prevent them of using DAS to share their data.

There was a need for a system capable of creating and hosting DAS servers that was simple enough to use so basic researchers could use it to share their data and make it publicly available.

Contributions

With the aim of removing as many obstacles as possible in the process of setting up a DAS server for researchers with interesting data in their hands, a new software platform has been developed: easyDAS.

easyDAS[20] is a hosted platform to automatically create DAS servers. Using a simple web interface the user can upload a data file -either GFF or any tabular text file-, annotate it and click a button and a new DAS server will be automatically created and made available to use by any DAS client. The annotation process includes all necessary information such as coordinate system selection and even an ontology browser based on OLS[21] to select a specific ontology term for every feature type, as recommended by the DAS specification. Using easyDAS, a user can create a new data source in a matter of minutes.

In addition, being a hosted solution, relieves the user from any responsibility regarding the maintenance of the server's hardware and software, and can be used to create data sources that won't change or disappear over time. In that regard, easyDAS could be a useful to distribute datasets linked to published papers, which are supposed to be reachable for a fair amount of time.

Built around Proserver[22], the perl DAS server, its back-end has been written in perl and the front-end has been implemented as a javascript web application using jQuery[23] to overcome cross-browser compatibility issues. Figure 1.5 shows an overview of the system architecture.

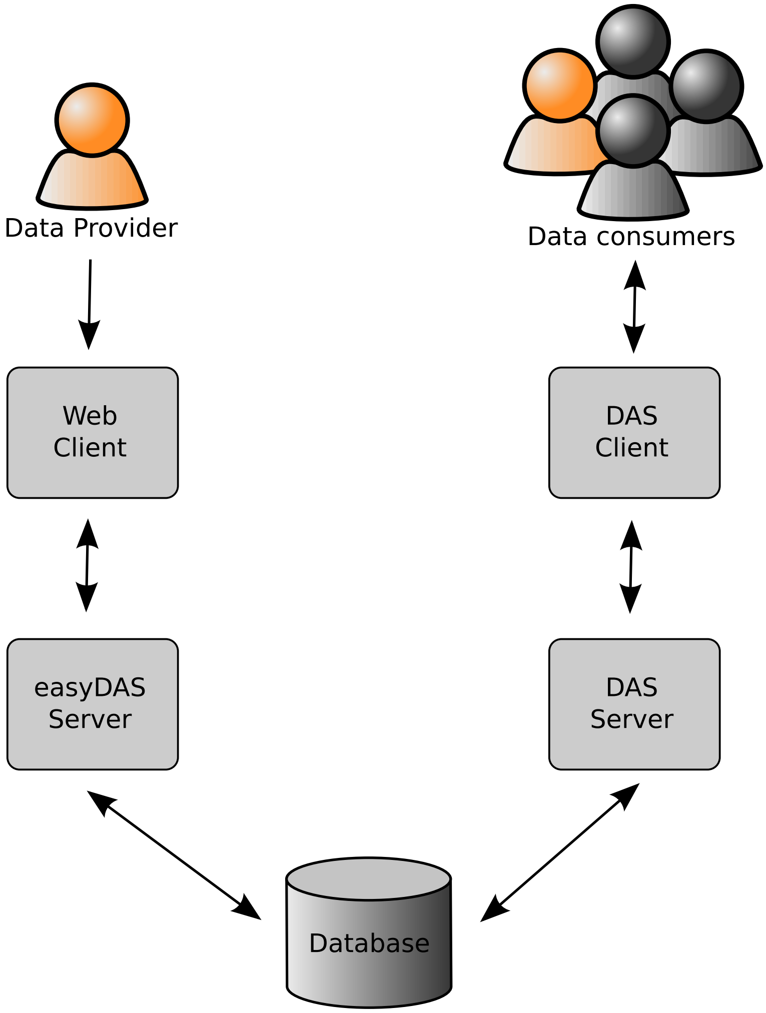


Figure 1.5 An overview of the easyDAS architecture. The data provider interacts with the `web client` to upload and annotate its data. The `easyDAS server` parses the uploaded data, stores it in the `database` and creates a new `DAS server`. Data consumers (including the original data provider) can access that data using any `DAS client`.

easyDAS was developed during a stay at the European Institute of Bioinformatics in the Proteomic Services team under the supervision of Andrew Jenkinson and Henning Hermjakob. An instance of easyDAS is running at the EBI and is freely available at http://www.ebi.ac.uk/panda-srv/easydas/ for anyone to use.

All easyDAS code has been released under the GNU LGPL license[24] and is available to download at http://code.google.com/p/easydas/. Being open source software, users are free to use it, modify it and redistribute it.

In addition to using the EBI's instance of easyDAS, it is possible to install custom instances of the software in other systems. An institution, for example, could set-up their own instance of easyDAS as a service for their research community to publish data using DAS -e.g. to share with collaborators, to view it in a genome browser, to link from publications, etc...- with minimal hassle and training required.

1.2.2 Visualisation

Motivation

Genome browsers provide a unique platform to browse, retrieve and analyze genomic data in an efficient and convenient way[25]. A number of genomic browsers exist, but three of them are by far the most widely used: Ensembl, UCSC, GBrowse and NCBI's MapViewer.

When this project was started, in 2007, all major genome browsers offered quite an static experience. It was certainly possible to browse and explore data, but is was done through a series of buttons moving the genome a certain amount of bases to left or right or zooming in and out. It was not possible, though, to use those gestures we have grown used to such as dragging the genome with the mouse or using the mouse wheel to zoom in and out fluidly.

The main architecture of the web based genome browsers was the same. They all had a relatively thin client-side part, mainly in charge of showing images and managing the interactions with the user and big back-end servers taking care of everything else, from accessing the data repositories to managing and integrating the data to creating the image representing it. For every change in the display parameters made by the user, a change in a zoom level of a displacement of the selected window by a few bases, triggered a request to the server that had to retrieve any data needed and create a new image to be displayed by the client side. This architecture has some advantages but imposes some restrictions to interactivity, since most user actions would trigger a network request with the associated response time.

An alternative approach where data rendering is performed in the client could help reduce network waiting times and produce a smoother interaction with the data representation, since once data had been transfered into the client, no more network requests would be needed and it would be the client itself that would redraw the genomic representation. At the time this project started, the power of the javascript engines in the web browsers and their graphical capabilities were limited. However, the presentation of Google Maps[26] the year before showing that it was possible to create a direct manipulation interface and the availability of the *canvas* element and it procedural drawing API, prompted us to investigate the feasibility of creating a highly interactive genome browser on the web and to investigate new interface paradigms for genome browsers.

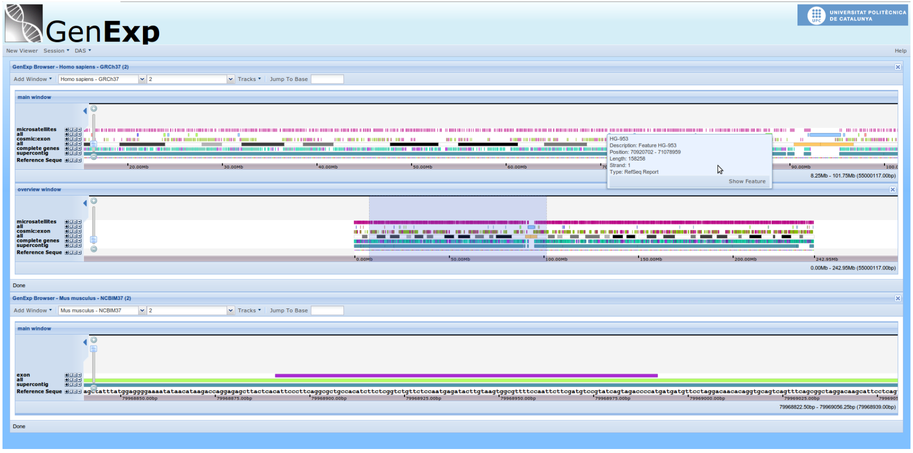
Contributions

Our aim was to investigate the feasibility of a genome browser with a high degree of interactivity which used the usual direct interaction conventions to manipulate the genome representation as we do with a mapping application such as Google Maps. To achieve that we decided to take advantage of the, new at the time, HTML5 canvas element[27] and its procedural drawing API. With it, it is possible to create bitmaps in the browser itself. Opposed to SVG or VML drawing, once the drawing is done, independently of the total number of features represented -one or one million- the browser only needs to manage one object to deal with the user moving the genome representation and so a very smooth experience can be achieved. Somehow, the canvas approach can be seen as a hybrid between the server created bitmaps and the SVG approach since everything is managed in the client side but the user interface code is really just showing a number of pre-drawn bitmaps.

So we created a new prototype genome browser called GenExp[28,29], an interactive browser with canvas based client side data rendering. It offers fluid direct interaction with the genome representation and it's possible to use the mouse drag it and use the mouse wheel to change the zoom level. GenExp offers also a number of quite unique features, such as its multi-window capabilities (Figure 1.6) that allows a user to create an arbitrary number of independent or linked genome windows in order to explore the same region of the genome with different parameters or a different region of a genome, even from a different species.

GenExp is also a DAS client and all data is retrieved from DAS sources. It allows adding any data source available including all data in Ensembl, UCSC and even the custom ones created with easyDAS or another DAS server. GenExp can be found at http://gralggen.lsi.upc.edu/recerca/genexp/.

The system is structured in two parts: a small back-end server -needed to overcome same origin policy (SOP) restrictions and providing some caching facilities- and the client. The light server has been implemented in perl and the front-end, with the data managing, drawing code and interface control, is entirely implemented in javascript using Prototype[30] as a base library and ExtJS[31] as the GUI and widgets library. All code has been released under the open source GPL license[32] and is available at http://code.google.com/p/genexp/



HG-953
Description: Feature HG-953
Position: 70920702 - 71078959
Length: 158258
Strand: 1
Type: RefSeq Report

agctatttatggaggggaaaatataacataagaccaggagagcttactcacattcccttcaggcgctgccacatcttctcggtctgttctccaatgagatacttgtaagtggcgttttccaattcttcgatgtccgtatcagtagaccccatgatgatgttcctaggacaacacaggtgcagtcagtttcagcggctaggacaagcattcctcag

Figure 1.6 Screenshot of GenExp showing three different genome viewers. The first viewer shows a region of the human chromosome 2, the second one is a linked view centered at the same base but with a broader zoom level and the third one shows also the chromosome 2 but from the mouse genome.

Additional Contributions to Visualisation

While DAS client libraries exist for perl -Bio::DAS::Lite[33]- and JAVA -JDAS[34], Dasobert[35]- no DAS client library existed for javascript. Explicitly dealing with DAS requests and responses -including any error condition-, parsing the XML results and transforming them into javascript objects required a fair bit of code and was needed for any web application that had to access a DAS source.

We developed a javascript DAS client library, jsDAS[36]. As the other software pieces presented jsDAS is open source software -in this case licensed under the LGPL- and is freely available at http://code.google.com/p/jsdas.

jsDAS is a complete DAS client library that will take care of everything DAS related in a javascript application, from request URL creation to the parsing of the responses and error condition notification. The library offers two API levels, one similar to the one offered by other libraries with an object acting as the proxy for a DAS source and another lower level one accepting DAS URLs, with both of them returning a javascript object representing the data received from the DAS source. In addition, since the library is highly modular, it is possible to use any of the modules independently either to build or parse DAS URLs, to parse DAS XML documents -e.g. if one wanted to access DAS XML documents in the local filesystem-, etc. jsDAS is javascript library agnostic -i.e. do not depends on jQuery, MooTools, Dojo...- and since it avoids namespace pollution -only a single jsDAS object is created- it can be used along them without any interference.

1.2.3 Integration

Altogether, the developed tools and technologies help in the whole process of publishing, sharing and integrating biological data. It is possible, for example, to use easyDAS to create a new DAS source with data from a given experiment, maybe making the source private so it does not appear in the public sources listings. Then, this source can be added to GenExp to study the characteristics of the dataset and specially its relation with other genomic features -such as genes, SNPs, fragile sites, other experiments...-. When a publication is derived from this work, a new public DAS source with a refined version of the original dataset can be created to make it available to anyone interested in it. Any researcher can then use it to include the data into a genomic browser -GenExp, Ensembl or any other accepting DAS- and use the data in their own research.

1.3 Structure of the thesis

The thesis is structured in two main parts: Dissemination and Visualisation. These parts are preceded by this introduction and followed by two chapters with the conclusions and a short follow up of the work presented here.

In the Introduction, the problems derived from the large amounts of data generated by biological laboratories and the challenges arising when trying to share and use and visualise it effectively are stated. A small section describing the exact challenges being addressed and the work done on each front follows, including a short description of the Distributed Annotation System (DAS).

Next is the part dedicated to biological data dissemination. The difficulties of efficiently disseminating biological data are explained, how making data publicly available is highly beneficial to the scientific community but only a few systems have been developed for a subset of selected data types. The Distributed Annotation System is presented along with its contribution to the sequence annotation data distribution. Since setting up the infrastructure to use DAS to distribute our own data might be also challenging for a large part of its target audience, easyDAS, a tool for the automatic creation of DAS servers we have developed is described in depth.

The second part is devoted to some aspects of biological data visualisation. After a short introduction of the broad field that data visualisation and specifically biological data visualisation is, we focus on visualisation of sequence annotations and genome browsers. After stating some of the user interaction limitations present in genome browsers, we present GenExp, a web based genome browser that uses client side data rendering techniques in order to offer a direct manipulation interface to the users.

Finally, the Conclusions chapter closes this thesis.

1.4 Open Source, Open Access and Open Data

I am a firm believer that collaboration is one of the the driving forces behind advances in science. Just as we are all expected to share our results via its publication in scientific journals to let the whole scientific community know about them and continue building upon them, I think that the more we share and disseminate our data, results and developments, the better for science as a whole.

Therefore:

- **Source Code:** All code implemented in the course of this thesis has been licensed under an open source license and is freely available via public code repositories.
- **Journal Papers:** All papers supporting this thesis have been published in open access journals or as open access papers in journals accepting both open access and closed access papers. Thanks to that, all of them are available free of charge to every researcher without the need of any journal subscription.

1.5 Supporting Publications

This thesis is supported by two main publications in scientific journals, a publication in the proceeding of a workshop and by a number of presentations in different national and internationl conferences. In addition, there are two highly related publications derived from the work developed in this thesis of which I am also an author. All published papers supporting this thesis has been published in open access journals.

1.5.1 Main Journal Papers

easyDAS

A paper published in *BMC Bioinformatics* in January 2011 presents easyDAS, a web based tool for the automatic creation of DAS servers. The paper describes the tool, its usage and has a brief description of the technical aspects involved in its development. The paper received a *Highly Accessed* badge due to the high number of views right after its publication.

Title: easyDAS: Automatic creation of DAS servers

Authors: Bernat Gel Moreno, Andrew M Jenkinson, Rafael C Jimenez, Xavier Messeguer Peypoch, Henning Hermjakob

Journal: BMC Bioinformatics

Publication Date: 18 January 2011

Notes: Highly Accessed paper

Full Citation: Gel Moreno B, Jenkinson AM, Jimenez RC, Messeguer Peypoch X, Hermjakob H. *easyDAS: Automatic creation of DAS servers*. BMC Bioinformatics. 2011;12(1):23.

Contributions by the authors: The original idea is from AMJ, RCJ and HH. Under the guidance from XM and with the helpful input from the two DAS experts AMJ and RCJ, I developed the idea and designed the system. The implementation and testing was done by me. I wrote the bulk of the manuscript, which was revised and approved by all the authors.

GenExp

The paper describing GenExp was published in July 2011 in *PLOS One*. The paper describes GenExp, a web-based genome browser using DAS data and client side rendering techniques. The text also includes a description of most of the challenges faced when developing a fat-client web-based genome browser and explains what approaches were used to overcome them in GenExp development.

Title: GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering

Authors: Bernat Gel Moreno, Xavier Messeguer Peypoch

Journal: PLOS ONE

Publication Date: 5 July 2011

Full Citation: Gel Moreno B, Messeguer Peypoch X. *GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering*. PLoS ONE. 2011; 6(7):e21270.

Contributions by the authors: XM and me conceived the project. With helpful suggestions and guidance from XM, I designed the system, implemented and tested it. The manuscript was written by me, and revised and approved by XM prior to publication.

GenExp Implementation

A previous paper about the implementation of GenExp was presented in the *2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008)* and the proceeding of the conference were published by Springer in the series *Advances in Soft Computing* in 2009.

Title: Implementing an Interactive Web-Based DAS Client

Authors: Bernat Gel Moreno, Xavier Messeguer Peypoch

Book Title: 2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008)

Series: Advances in Soft Computing

Publisher: Springer

Publication Date: 2009

Full Citation: Gel B, Messeguer X. *Implementing an Interactive Web-Based DAS Client*. In: Corchado J, De Paz J, Rocha M, Fernández Riverola F, eds. 2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008).Vol 49. Springer Berlin / Heidelberg; 2009:83-91.

Contributions by the authors: As a preliminary publication based on the same project as GenExp, the contributions by the authors are the same. XM and me conceived the project. I designed the system, implemented and tested it under the supervision of XM and I wrote the manuscript, which was revised and approved by XM prior to publication.

1.5.2 Additional Journal Papers

Dasty3

Dasty3 is a protein oriented DAS browser developed at the EBI and used both as a standalone application and as part of other applications such as Uniprot. Dasty3 is the evolution of Dasty2 and as it, it is completely web based. The newest incarnation of Dasty abstracted the DAS communication layer out and used jsDAS, our javascript DAS client library. I participated in the design phase of the project, where the whole widget based architecture was defined, and wrote most of jsDAS. Dasty3 was presented in an applications note published in *Bioinformatics* in July 2011. The full paper is available in the appendix .0.2.

Title: Dasty3, a WEB framework for DAS

Authors: Jose M. Villaveces, Rafael C. Jimenez, Leyla J. Garcia, Gustavo A. Salazar, Bernat Gel, Nicola Mulder, Maria Martin, Alexander Garcia and Henning Hermjakob

Journal: Bioinformatics

Publication Date: 28 July 2011

Full Citation: Villaveces JM, Jimenez RC, Garcia LJ, Salazar GA, Gel B, Mulder N, Martin M, Garcia A, Hermjakob H. *Dasty3, a WEB framework for DAS*. Bioinformatics (Oxford, England). 2011; 27(18):2616-7.

Contributions by the authors: This project was mainly lead by JMV, who implemented and tested the system. Ths design of the system and the base infrastructure that later became BioJS was devised and designed by RCJ, LJG, GAS and me. In addition, I conceived, designed and implemented JsDAS, the javascript DAS client library powering the data access layer of Dasty3. NM, MM, AG and HH supervised the project.

MyKaryoview

MyKaryoView is an integrative browser specialized in data from direct-to-consumer (DTC) genetic testing providers such as 23andMe. It is mainly aimed at what is known as Do It Yourself bioinformatics where the people getting their genotype data has the sufficient knowledge and expertise to want to analyze their genotype data by themselves and integrate it with other data sources. MyKaryoView uses easyDAS as the the way of creating DAS sources from the original data files provided by the DTC companies. MyKaryoView was presented in a paper published in *PLOS One* in October 2011. The full paper is available in the appendix .0.2.

Title: myKaryoView: A Light-Weight Client for Visualization of Genomic Data

Authors: Rafael C. Jimenez, Gustavo A. Salazar, Bernat Gel, Joaquin Dopazo, Nicola Mulder, Manuel Corpas

Journal: PLOS ONE

Publication Date: 26 October 2011

Full Citation: Jimenez RC, Salazar GA, Gel B, Dopazo J, Mulder N, Corpas M. *myKaryoView: a light-weight client for visualisation of genomic data*. PLoS one. 2011;6(10):e26345.

Contributions by the authors: This project was lead by RCJ, who coceived, designed and implemented the project together with MC. GAS offered technical help wih respect to the DAS writeback and I helped in the integration of the system with a customized version of easyDAS. JD, NM and MM supervised the project.

1.5.3 Research Interests

Bioinformatics is a research field at the interface between Biology and Computer Science. While I come from the computational side and my PhD thesis has been done at the Software department of the UPC, I've been drifting closer to the biological side for some years and so my research activities have changed quite a lot (See the complete CV at Appendix .0.2).

Since even before the beginning of my PhD I was collaborating as a biostatistician with the Molecular Oncology group at the department of Human Anatomy of the UB, led by Dr. Mariano Monzó. We started with simple statistical analysis of clinical and molecular data to identify prognostic factors for different cancer types as well as predictors for treatment response, but over the years the projects evolved to include new technologies like microarrays and new discoveries such as microRNAs and so did the analysis, moving from classical biostatistics to bioinformatics data analysis. My final degree project was linked to my work there and tried to study how the microRNAs encoded by some DNA viruses may act as regulators of host gene expression, with emphasis in the Epstein-Barr virus. I've been collaborating with Dr. Monzó's group for more than 6 years and a number of publications resulted from that work, including the first microRNA profiling of classic Hodgkin's Lymphoma[37] and the identification of mir-34a as a prognostic factor for Non-small-cell Lung Cancer (NSCLC)[38].

In 2011, when most of the work for my PhD was done and I started writing this thesis, I moved to the Genetic Variation and Cancer group, leaded by Dr. Eduard Serra, at the Institute of Predictive and Personalised Medicine of Cancer (IMPPC). For almost three years now I've been working in the study of hereditary cancer and specially in malignant tumors affecting Neurofibromatosis Type-1 patients: Malignant Peripheral Nerve Sheath Tumors (MPNST). Applying the most recent sequencing technologies and an integrative bioinformatic analysis we are identifying genes driving the progression from benign neurofibromas to malignant MPNSTs and trying to better understand their biology. In addition, we are collaborating with Dr. Conxi Lázaro's hereditary cancer group at ICO in establishing and validating a mouse xenograft model to perform pharmacogenomic studies and with the Hereditary Cancer Genetic Diagnostics group with Dr. Elisabeth Castellanos in designing and establishing new diagnostics procedures based in high throughput sequencing. A couple of papers have been published from this work[39,40].

From my first formal education in computer science to my current research, mainly cancer genomics, I learned a lot and my interests evolved and changed, but it was certainly worth it.

Part I

Dissemination

Chapter 2

Biological data dissemination and integration

For many years the dissemination and integration of biological data has been an open problem for bioinformatics. It was 1985, before the World Wide Web, when in the National Academy of Sciences Report titled "Models for Biomedical Research: A New Perspective"[41], the Committee on Models for Biomedical Research stated that *"new generalizations and higher order biological laws are being approached but may be obscured by the simple mass of data"* and proposed the creation of a *"Matrix of Biological Knowledge"*, an *"encyclopedia of biological knowledge"* that is *"extensively cross-referenced"*. While today the idea of creating a single unified resource with all the biological data and knowledge ever produced would be deemed unrealistic, in the last two decades significant efforts have been made to create resources linking or integrating multiples sources of biological data and to create comprehensive databases for narrow fields.

Today the challenges of distributing and integrating biological data and knowledge have not yet been solved but important advances has been made and new approaches and technologies are being developed. Generalized access to the internet has solved one of the major technical problems, the actual data distribution, but it is clear that the accessibility alone is not enough. Another technology, the semantic web holds the promise of virtually creating that *"Matrix of Biological Knowledge"* but it seems to be one these revolutions always "a couple of years away".

2.1 Data dissemination

Data and results derived from experiments -either *in silico* or *in vitro*- should be made available to the research community in a format that is useful and easy to be used. This affects a number of aspects of the process: standard formats to ensure interoperability, sufficient metadata about the experiment and samples, easy access and possibility of sharing with peers. The idea is that data and results flowing freely all through the research community will have benefits for those producing data, those using it and the community as a whole.

For many years, scientific journal papers has been the main distribution channel of scientific results and raw data. For an experiment involving, for example, five magnitudes mesured in five different conditions, a journal paper is a good platform to present, explain and disseminate the results: it is possible to create a few tables and graphs representing the generated data and any conclusion can be explained thoroughly. However, if the experiment consists in measuring a magnitude for tens of thousands of entities over a tens or hundreds of samples -as in expression microarrays or in SNP arrays, for example- it is evident that a journal paper is not the most efficient way to disseminate the data and results. While a journal paper might be well suited to explain and discuss the conclusions and their implications, raw data and even results from high throughput experiments in this era of *-omics* and *in-silico* analyses make traditional ad hoc methods of publishing and sharing data impractical[42]. Performing such an experiment and reporting only the 10 most different entities between two conditions -e.g. most differentially expressed genes, or differentially methylated CpG islands- instead of releasing the whole dataset and the complete results list might be limiting. Limiting the published results to a few top scoring entities leaves many potentially interesting results hidden and without the actual raw data is not possible to reanalyze the dataset using different approaches nor including them in any kind of meta-analysis.

Regarding the storage and distribution of the biological data there are two main approaches: centralized, with a few central services are used to concentrate all data generated by anyone and take the responsibility of storing and distributing it, or distributed, where these functions are performed by the researchers generating the data. Each approach has it advantages and drawbacks and both are widely used nowadays for different kinds of data.

2.1.1 Big Repositories

Big data repositories have been a very valuable tool for bioinformatics. Usually provided by key institutions like NCBI or EBI-EMBL, they are free centralized storage and distribution hubs and are usually available to any researcher to use.

In general, data repositories are designed to accept a single type of data or at most a few tightly related ones -i.e. different types of microarray experiments:

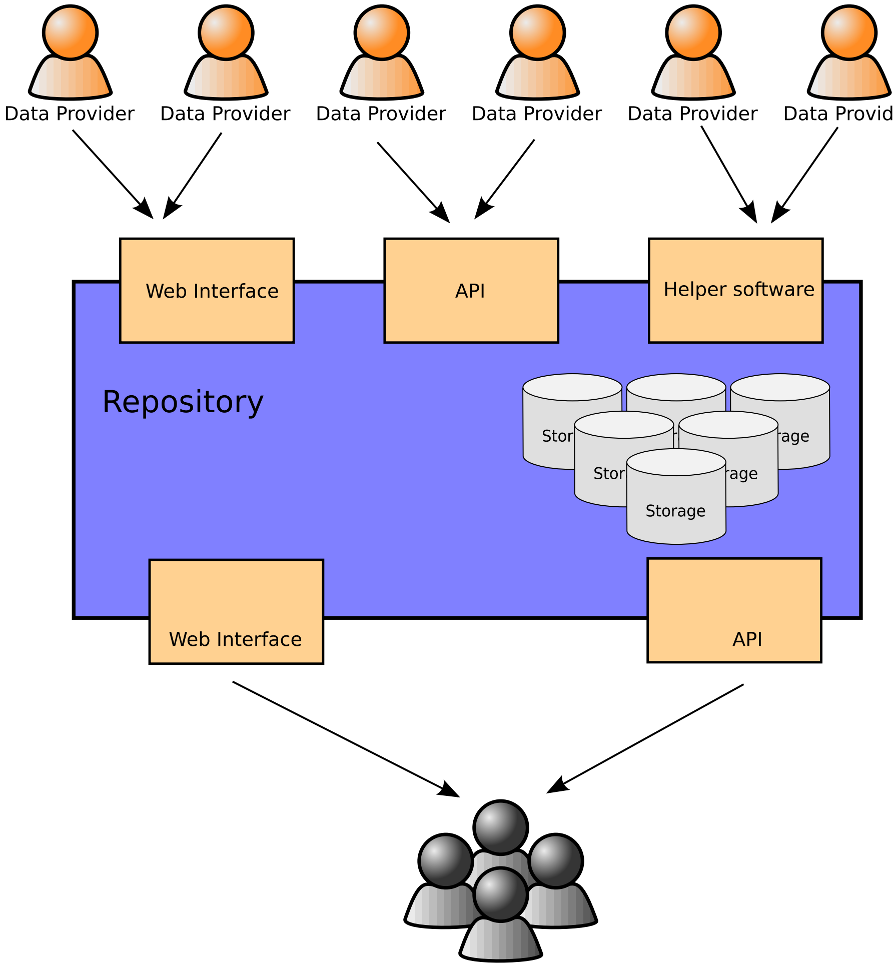


Figure 2.1 The centralized model. In the centralized model all data providers contribute data to one single data repository using any of the available contribution methods. The central repository stores and manages the data and offers one or more data access interfaces, which are used by the final users.

gene expression, aCGH...- and are highly optimized to manage that type of data. Many of them offer easy to use interfaces to upload and retrieve data and some even perform a minimal and standard preliminary analysis of the uploaded datasets (Figure 2.1).

The Gene Expression Omnibus (GEO)[5,43] is an example of a highly successful data repository. It started as a repository for high-throughput gene expression microarray data but has evolved to include any experiment based on microarrays -gene expression, methylation, aCGH, protein arrays, ChIP-on-chip-

and even some based on next-generation sequencing[44]. Since its inception, GEO -and the other main repository for microarray data, ArrayExpress[45,6], have become the main source of microarray data. Researchers uploading data into these repositories are encouraged to provide enough metadata about their experiments following the MIAME guidelines (Minimum Information About a Microarray Experiment)[46,47]. This abundance of metadata, has been a very helpful to the many statistical and computational groups developing new analysis strategies for microarray data as well as to the groups performing meta-analysis on that data.

If a centralized repository is available for a certain data type, it is a powerful and convenient way of distributing data. They are, in general, easy to use -with many of them having helper applications and documentation guiding you through the process of submitting data- and require a minimum of metadata about the datasets being submitted. In many cases a big institution is backing them, increasing the probability of the project having a long life and so data being accessible in the foreseeable future. The repositories have usually well planned backup policies and the users do not need to take any responsibility on data maintenance once it has been uploaded nor have to pay for any of the associated costs such as storage and bandwidth.

However, centralized repositories have their own drawbacks. They are usually highly specialized and mainly targeted at storing raw or lightly preprocessed data and not final results -or at least not in a general level- and some have strict rules on the metadata, ensuring a minimum of information about the datasets. The researcher have a little or none control over how the data will be distributed and the available interfaces to access to it. While central repositories might be quite efficient in terms of cost per dataset they represent a financial effort for the institutions maintaining them mainly due to the huge and ever increasing volume of data stored (Figure 2.2). Probably, then, a central repository is only a good approach for a certain data type when the amount of data is important and when it will be backed by an institution with the sufficient economical strength.

2.1.2 Distributed model

An alternative to the big repositories for datasets or results not fitting in any of them or for those who need extra control over how their data is represented, stored or distributed, is the distributed model, with a number of small systems serving the data. Basically every researcher, research group or institution sets up their own data server and decides on what they make available to the world and the data consumer will have to somehow find the resource, access to it and get the data in a usable form. This approach makes publishing data of any kind viable but presents problems with regard to discovery and integration (Figure 2.3).

Using a distributed model, data providers retain absolute control over what exactly is shared -raw data, preprocessed data, results, everything-, with whom it will be shared -fine grained user access control is possible- and how the data is annotated, deciding on the amount of metadata available and its format. In addi-

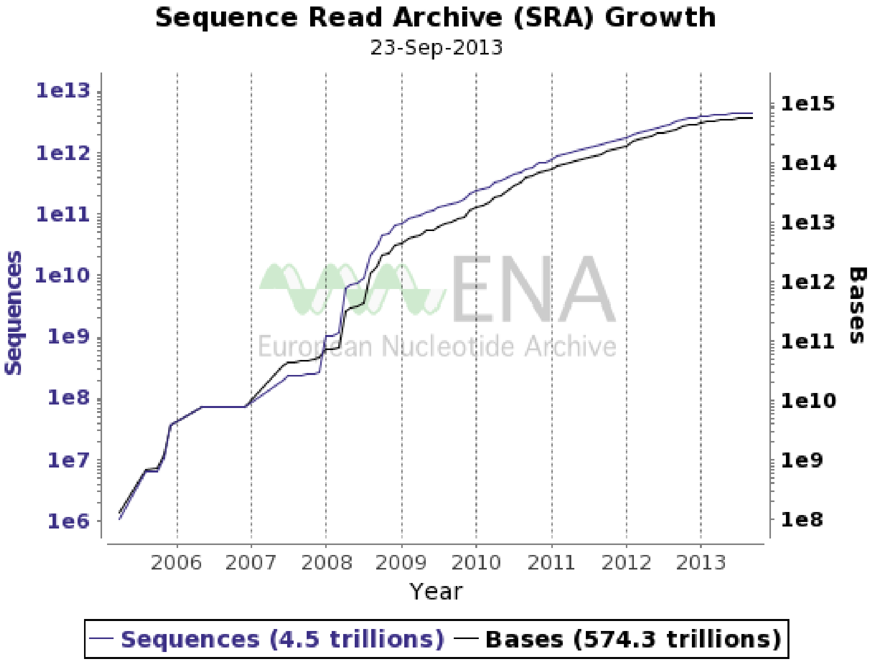


Figure 2.2 Sequence Read Archive data growth. The Sequence Read Archive (SRA) at the European Nucleotide Archive (ENA)[48] is an example of central repository for sequencing data. This chart shows the growth rate of the amount of data stored in the database in logarithmic scale.

tion, custom interfaces can be used to present the data in the most convenient or meaningful way or standard formats can be used. Data can be changed, modified and updated as needed. Furthermore, the total cost is effectively distributed and the cost of setting up and maintaining a small server is low and could be afforded by many.

On the other hand, all that publishing freedom comes at a cost. There is no minimum of metadata guaranteed, no quality checks enforced and the data might not be usable out-of-the-box and might be in non-standard formats. Although mitigation by resource federation is possible, finding the right data sources might be difficult and some of them might not be used simply because they are not known. The cost of setting up, managing and maintaining a server is low but not null and the aggregated cost of all the small data sources is probably higher than that of a central repository and since usually these kind of services are usually tied to a certain project or grant, there's no guarantee that data will be available once the project has finished. Finally, setting up and maintaining a server is not trivial and many research groups lack the expertise to do so.

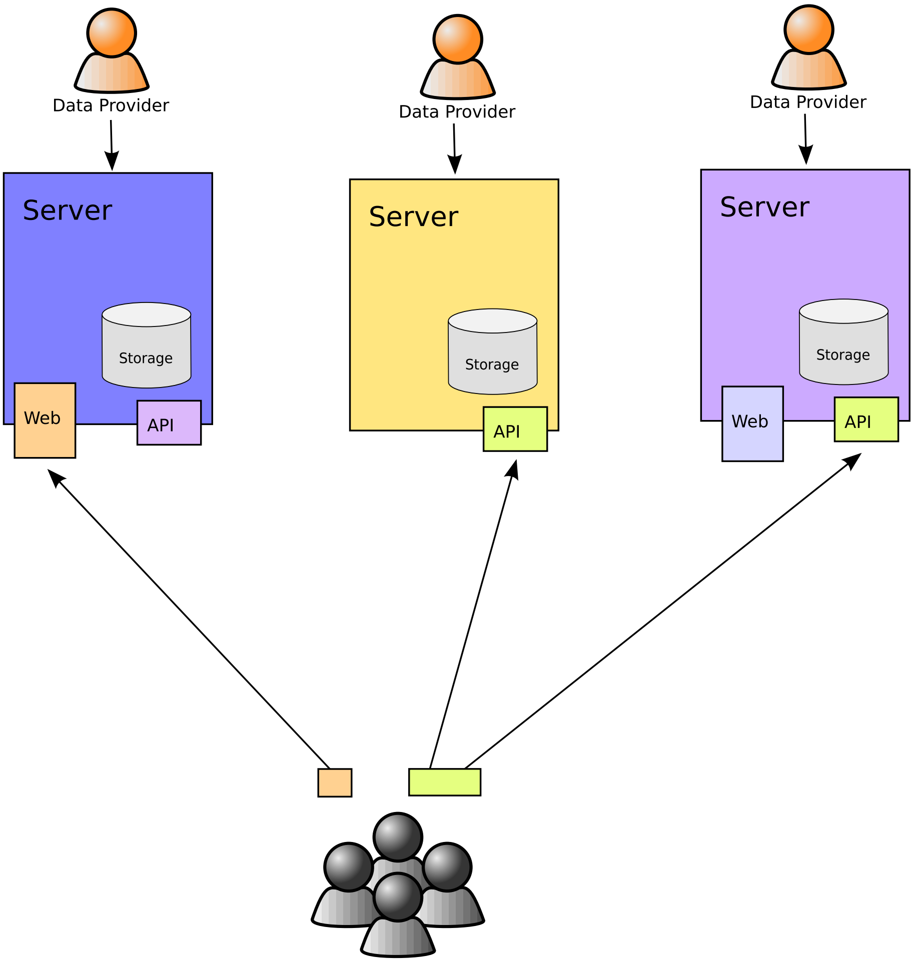


Figure 2.3 The distributed model. In the distributed model each data provider stores and manages its own data and offers one or more data access interfaces. The end users directly connect to the data provider interfaces to extract the data.

2.2 Genomic and Protein Annotations

General biological data dissemination and specially integration is an open problem and has been an active research field for the last quarter of century. It is not the aim of this thesis to solve the general problem but to focus on dissemination and integration of genomic and protein annotations.

2.2.1 Annotations

In general, an annotation is a piece of data linked to something to provide addition information. From underlining and taking notes on a text book to identifying people in a picture or adding a comment to a youtube video, many forms of annotations are used in our daily life.

In computation biology, annotations mainly refer to the placement of features on biological sequences. The position of a gene and its exons in a chromosome, the position of an alpha-helix in a protein, a microRNA target region in an mRNA or even the GC content over a genome are annotations on biological sequences. In addition to the "intrinsic" features of the biological sequences, experimental results can also be seen as annotations -e.g. the position and shape of a ChIP-seq peak, the number of reads per base derived from an RNA-seq experiment, the mutations found when sequencing a gene- and might be treated in general as such. In general, only parts of the sequences are covered by an annotation, but it is perfectly possible to annotate the entity itself for example adding functional annotations to a protein sequence, or bibliographic references where a given sequence was studied.

In any case, however, an annotation will always be tied to an entity being annotated and will not make any sense by itself. The position at which somebody's face is located in a picture does not make any sense without the picture itself, as having the position of a domain in a protein does not make any sense without the protein itself. The concept of the reference object, the entity being annotated is fundamental and it is very important that annotations are linked to their reference object.

2.2.2 Reference sequences

In our case, the reference objects being annotated are sequences, either genomic or protein. An important point to take into account is that for annotations to make sense it is necessary to tie them to the correct reference object and so we need to uniquely identify the reference sequences. For example, we cannot reference an annotation to a sequence identified as *the human chromosome 17*, since different versions of the genome, different assemblies, have minor differences in the genomic sequence and the chromosomes sequences change and the position of the annotations should change accordingly.

When creating unique sequence identifiers, different approaches have been taken for different sequence types. In the case of genomic sequences, the establishment of commonly accepted authorities and a release cycle based on named and stable assemblies as opposed to an ever-evolving assembly was good enough for many years. Recently, , an international consortium, the *Genome Reference Consortium* (GRC)[49], was established to be the only authority releasing reference assemblies and the release cycle was changed to maintain the reference genome stable for much longer -e.g. as of September 2012, the current assembly is GRCh37, released in March 2009 and a new release, GRCh38, has been scheduled for summer 2013, making it a 4 year cycle- adding minor releases, called patches -with all changes introduced thoroughly annotated-, to keep the genome up to date without changing the main reference.

Chapter 3

The Distributed Annotation System

The Distributed Annotation System (DAS) is a protocol designed to publish and integrate annotations on biological entities, mostly sequences, in a distributed fashion. DAS is structured as a client-server system where the end user uses a client to retrieve data from one or more servers and merge, process and visualize it. DAS development was started in 2001 by WormBase[12] as a way to distribute and share its genomic annotations among its users and was soon adopted by the Ensembl project[13] and other databases. Currently a central registry, the DAS Registry[14], has more than 1600 data sources registered and these sources have been created by more than 50 institutions.

The idea behind DAS is that many different laboratories and groups might have relevant but partial knowledge regarding a biological entity, such as a genomic sequence. To combine, visualize and explore all this data jointly, it should not be necessary to merge it all in a central database but it should be possible to merge it *on-the-fly* by the software being used by the end user. This approach has some interesting advantages such as being highly dynamical with respect to the set of data sources being used -i.e. the end user could be able to select any compatible source, including those created by himself, to be added to the view instead of having to ask to the hypothetical central database to add that data and make it available. With data being under control of those producing it, new releases, updates and corrections can happen more often, without being tied to the common release schedule of a central database.

3.1 A short history of DAS

The Distributed Annotation System was initially developed by WormBase[12] as a way to distribute and share its genomic annotations. It was rapidly adopted by the Ensembl project[13] who used it to distribute its own data setting up a DAS server as well as a mean for users to include additional data into the Ensembl Genome Browser which acts as a DAS client. It was in 2001 when the first DAS specific paper, by Dowell *et al.* came out[50], presenting the version 1.0 of the protocol. In its initial version, the DAS system already had most of its keys features and characteristics. There was the concept of servers and sources, the annotation sources and reference servers, the features types and the concept of entry points to access the sequences. Stylesheets were also described as well as the use of HTTP headers to report the response status.

The first version of the protocol, however, was heavily influenced by the GFF format and most of the fields describing a feature are exactly the fields of a GFF file. That initial version, in addition, was specifically designed to annotate genomic sequences. As an example, the command to retrieve the underlying sequence being annotated was `dna`, ignoring the possibility of aminoacid protein sequences. In contrast, the current command to retrieve a sequence is `sequence`, taking a more generic approach. Some of that bias onto genomic features is still present on the protocol, for example in the form of an optional feature attribute `phase`, meaningless in most of the protein annotations.

Software tools implementing the standard both in Perl and Java were already described in the initial paper. Two servers were presented: Dazzle, written in Java, and another one to transform ACeDB databases[51] into DAS servers, written in Perl. Two clients were also presented, Geodesic, a Java standalone client and DasView, a server side web client written in Perl.

The eFamily project[52] had as its main aim to integrate the protein annotation data (either structural or based on sequence analysis) and decided to create a network of independent databases instead of a centralized repository and chose to use DAS as their standard data interchange format. At the same time, the BioSapiens Network of Excellence adopted DAS as the mechanism of sharing proteomics data among member institutions[53][54][55]. The DAS protocol was extended to support protein annotation data while it maintained a broad backward compatibility. Two new commands were added, `alignment` and `structure`, and a new rich standalone protein oriented, SPICE, was presented. In addition, a new service, the DAS Registry[56], a central listing for public DAS sources, was developed. These extensions were published in two papers by Prlić *et al.* in 2005[57] and 2007[14].

While the extensions made to the DAS protocol until then were mostly backwards compatible, in 2007 a project that aimed to define a completely new standard for DAS was concluded. The new specification was named DAS2 and was published on the DAS website (http://biodas.org/documents/das2/das2_protocol.html). DAS2 was an ambitious project that added new interesting features and concepts to DAS such as an extensive use of ontologies to semantically annotate every piece of data, or URIs to uniquely identify features. The writeback defined how data could flow from the user to the servers and explored the concept of distributed and collaborative annotation. While very well engineered and technically sound, the DAS2 specification was not backwards compatible and much more complex than version 1.5. It actually redefined most of the existing commands and concepts and rendered most of the previous infrastructure unusable.

For some time there was an big controversy within the DAS community about whether the new version should be adopted or the existing services maintained. While some sources decided to adopt the new version, most of them remained on the older version. During the 2009 DAS Workshop held on the Genome Campus, Hinxton, UK, it was generally agreed that the most useful additions from DAS2 would be gradually back-ported to DAS1 and that the work by the community would be centered on the evolution of DAS1.

Finally, in 2008, Jenkinson *et al.* published a new paper[42] presenting the specification 1.53 of the protocol and consolidating the specification. A lot of the additions presented there had been in use for some years, and some even were inspired by the DAS2 project. While the aim of that paper was to publish an "official new specification for DAS consolidating on the specification what was actually used by the most important clients and servers, it also incorporated new ideas developed during the previous years. Five additional commands were added and an ontology for protein features was presented. In addition, a new server-side data preparation option (binning) was defined and additional options to stylesheets were added.

On September 2010 the DAS 1.6 specification was presented on the projects mailing list and website and is, since then, the current version of the DAS specification. Following the idea of gradually improving the existing specification, some commands have been deprecated and the XML formats have been slightly modified to better accommodate new data types. As for now, version 1.7 is in the works and a public extension proposal program in place to get new ideas from the DAS community. It is expected that version 1.7 will incorporate some of the ideas of DAS2 back into DAS1, such as the writeback, and will be a new milestone on the evolution of the DAS protocol.

3.2 Architecture overview

The Distributed Annotation System has a client-server architecture based on web standards. The servers are the data providers, offering access to one or more data types, and the clients, the software used by the end user, connecting to them via standard HTTP requests. Servers have a variable number of REST endpoints serving different types of data -sequences, features (sequence annotations)- and metadata -data types available, description of the data source- and responses use a custom and well defined document format based on XML (Figure 3.1).

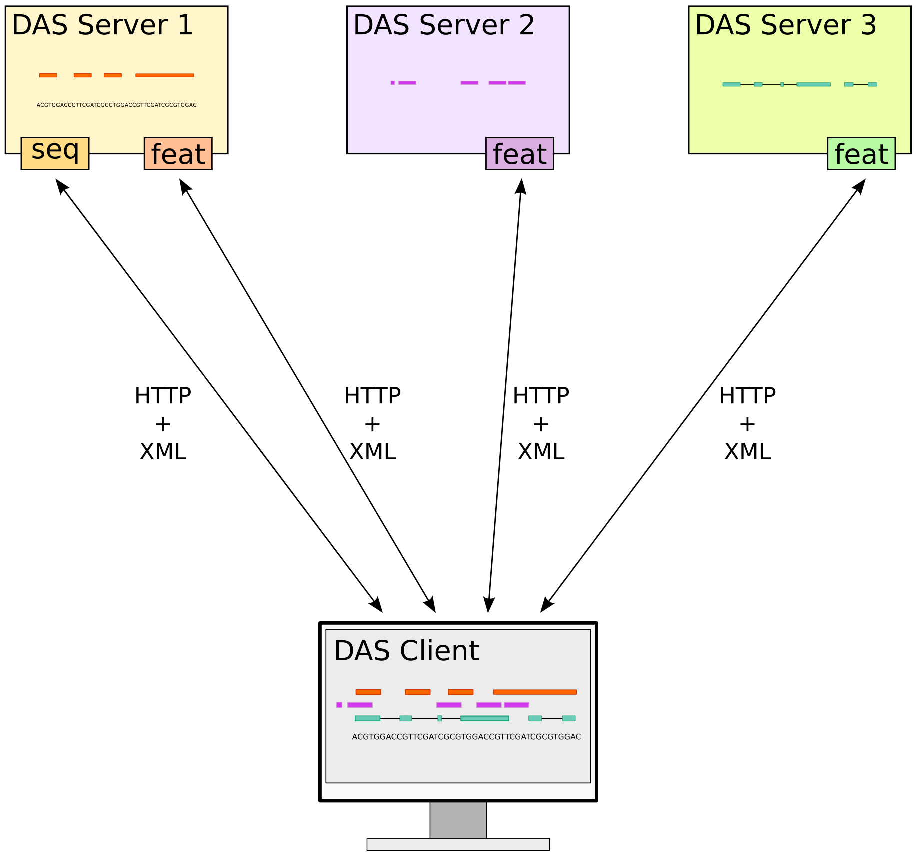


Figure 3.1 Overview of the DAS architecture. A number of DAS servers exist and each of them implement a subset of the DAS commands (sequence, features...). A client connects to different DAS servers to retrieve the desired information using a combination of HTTP + XML. Data integration and representation is performed locally by the client.

DAS has a *"dumb server, clever client"* architecture, which holds a number of advantages. For example, the minimal resources and time required of data providers to expose their data means more sources can be integrated and more readily[42]. However, the *"clever client"* part of the architecture means that DAS clients need to take care of data management and integration in addition to any processing, visualization and user interaction needed.

3.2.1 Servers and Sources

The concept of *Server* and *Source* in DAS is quite specific and often problematic for DAS novices. A server is a unique URL serving data on the DAS network. Although in many cases it will be a set of instances running behind a load balancer, it can be seen as an instance of a server software running on a specific physical machine.

A server provides one or more sources, which are the logical endpoints where DAS queries are sent. Each of this sources will provide a specific set of related data -e.g. data from the same organism, of the same type, etc-.

For example an institution could have a central DAS server at http://institution.org/das/ and have a separate source for each organism at

- http://institution.org/das/organism1
- http://institution.org/das/organism2
- ...
- http://institution.org/das/organismN

or for data published on different papers at

- http://institution.org/das/paper1
- http://institution.org/das/paper2
- ...
- http://institution.org/das/paperN

This two tier organization of data sources is usually used to introduce structure and organization to DAS URL naming.

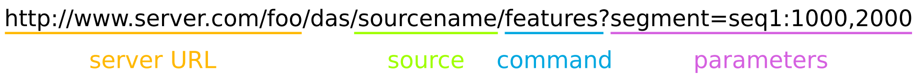

http://www.server.com/foo/das/sourcename/features?segment=seq1:1000,2000
server URL
source
command
parameters

Figure 3.2 DAS request example. An example of a DAS request with its different parts marked in different colors. This request to the source sourcename of the server http://www.server.com/foo will retrieve all annotation -features- overlapping the region from position 1000 to 2000 of the biological sequence identified by seq1.

3.2.2 DAS requests

As in every REST service, DAS requests are standard HTTP URLs. They encode the name of the server, the data source to be queried, the command and any needed parameters.

Figure 3.2 shows an example of a prototypical DAS request. In this example the DAS server is accessible at the URL http://www.server.com/foo and has at least one source, the one we are querying, named sourcename. The command being issued is features, that will retrieve a list of annotations and we are requesting it to return all the features overlapping a definite region: from position 1000 to 2000 in the sequence identified named seq1.

A list of the available commands and its parameters is can be found in Section 3.3.6.

In addition to the parameters included in the URL, some additional options can be specified as custom the HTTP headers. These custom options are identified with the prefix X-DAS and include X-DAS-Version to specify the version of the DAS protocol in use and X-DAS-Client to refer to the DAS client used. A complete description of the available headers can be found at the DAS specification.

3.2.3 DAS Responses

The responses to DAS requests are XML documents in a DAS specific format. The specifics of the XML format depend on the DAS command used but all of them are similar and quite simple. Figure 3.3 shows an example of a DAS XML response. The structure of the document is clear, simple and mostly human-readable. In this case we can see that the sequence requested -positions 32890598 to 32890664 of sequence identified as 13- is atgcctattggatccaaagagaggccaacattttttgaaatttttaagacacgctgcaacaaagcag.

As with DAS requests, HTTP headers are used to add additional information, including X-DAS-Version, X-DAS-Status similar to the HTTP status or a more complex X-DAS-Capabilities describing the parts of the protocol the server implements.

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="/das/das.xsl"?>
<!DOCTYPE DASSEQUENCE SYS-
TEM "http://www.biodas.org/dtd/dassequence.dtd">
<DASSEQUENCE>
<SEQUENCE id="13" start="32890598" stop="32890664" version="1.0">
atgcctattggatccaaagagaggccaacattttttgaaatttttaagacacgctgcaac
aaagcag
</SEQUENCE>
</DASSEQUENCE>
```

Figure 3.3 Example of a DAS response. In this case is a response to a sequence request

3.2.4 The DAS Registry

The DAS registry[14] a web-site and a set of web-services that informs the clients about the available sources. Any DAS source owner is free to register it at the DAS registry and will be made available to the DAS clients. Clients can programmatically retrieve a list of the available DAS sources meeting a set of conditions -e.g. for a given organism, annotating a certain genome version, with a specific data type...- and the registry will answer with all the information needed to send requests to them. In addition, through its web site, it is possible for a user to manually search and explore the list of available sources.

In addition, The DAS Registry periodically performs a validation of the DAS sources to check it status and its conformance to the DAS specification. If any non-conformance is found, it is reported to the owner and to the potential users of the source.

Lately, a new option to automatically register new sources via the `sources` DAS command has been implemented, opening the door to auto-registration of DAS sources.

3.2.5 DAS is not SOAP

While DAS and SOAP share the same philosophy of distributed and loosely coupled service and data providers and even share some of the underlying technologies: HTTP, XML, etc, DAS and SOAP are not the same and have some important differences.

From a technical point of view, DAS is based on the REST architecture that uses URLs to identify the documents and queries as opposed to the full fledged XML document needed to perform a SOAP query. This decision has some important implications, such as the ability to use the browser to interact with a DAS server.

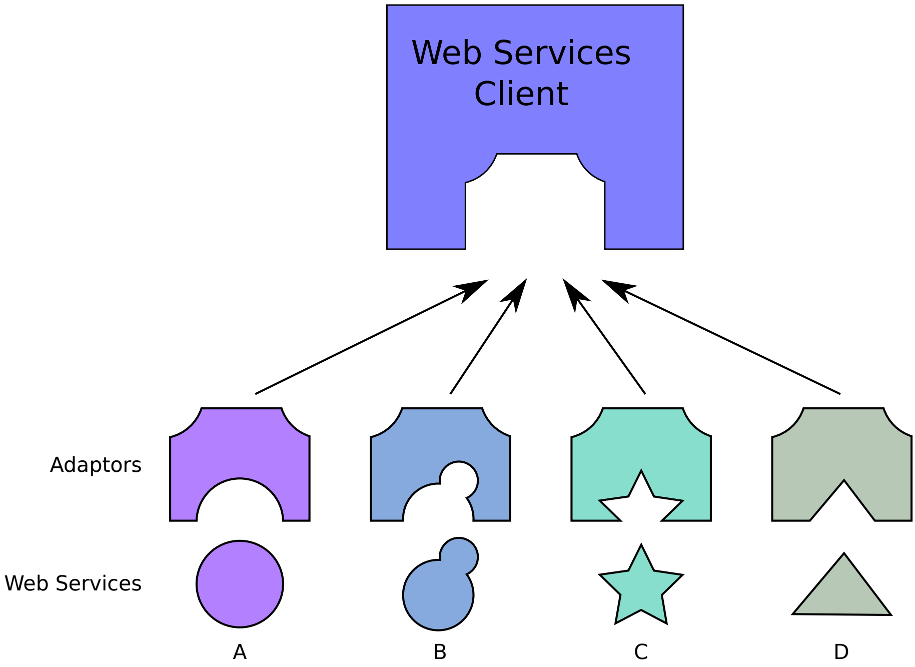


Figure 3.4 Web Services. Each Web Service defines its own interface. To access data from different web services, a client needs to include specific adaptor code to access the different web services.

The usual addition of XSLT transformation documents to the DAS responses further encourages this direct interaction of the user with the DAS server making DAS responses human readable. In addition, all DAS sources share exactly the same interface (or at least a well defined subset of it) is a key features of DAS. This standardization greatly facilitates client development by removing the need of adaptors and specific end-points for every new webservice added, and actually, new data sources are usually added to the clients automatically without any intervention by the user (*see figures 3.4 and 3.5*).

The most important differences, though, lie on the definition and organization of the DAS services. A standardized semantic layer on top of the syntactic standardization is another important feature of DAS. This ensures that clients al servers are talking about the concepts when serving and requesting data. Predefined semantics are very important for data integration and further facilitate the automatic addition of new data sources into clients. Predefined semantics limit the expressiveness of the data formats but the extended integration capabilities have been worth it.

Another difference is that DAS data sources are limited to serving static data and, in theory, can not run any calculations in order to return the responses to

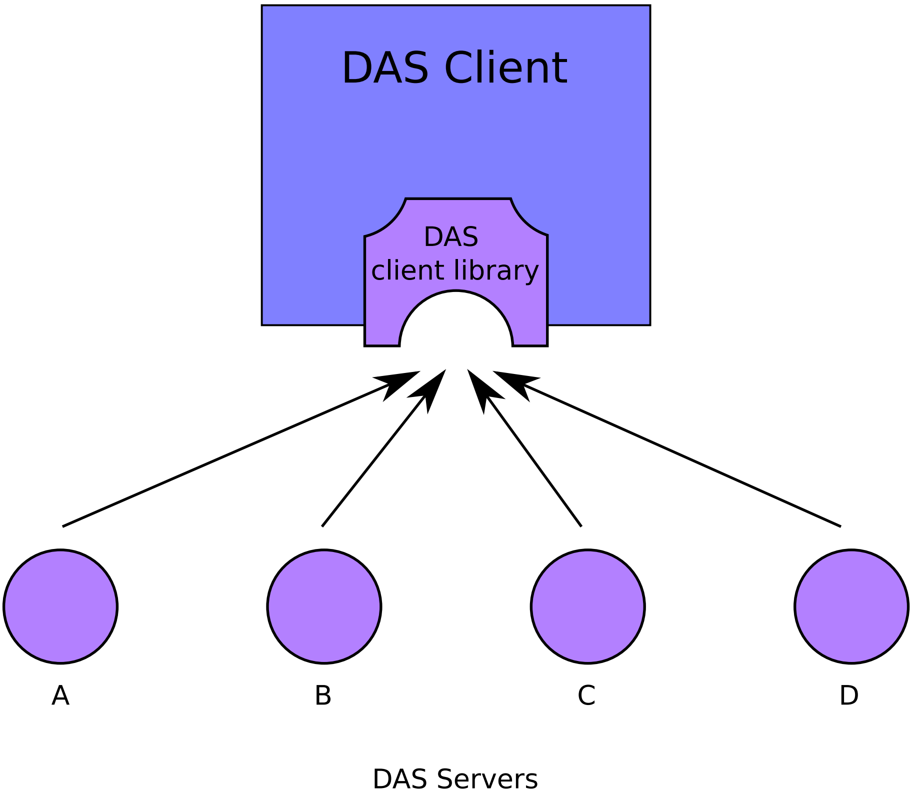


Figure 3.5 DAS. All DAS servers implement the same standardized interface. DAS clients need a single adaptor (the DAS client library) to access all data in any DAS server.

the queries. Response time is recommended to be less than 10 seconds although many sources, specially those with dense data, may take way longer than that in returning the response to a query requesting information about a long genomic range.

3.3 Important concepts in DAS

There are a few concepts with a specific meaning when dealing with DAS that need to be known in order to work with it. The most important are briefly explained below.

3.3.1 Coordinate systems

The Distributed Annotation System is used to publish and retrieve annotations of features on biological sequences and can describe the exact position of the feature on the sequence and its nature. However, an important question arises: which sequence exactly is being annotated? One could assume that simple answers to this could be "The aminoacid sequence of the protein derived from the gene BRCA1", but which one of the twenty seven isoforms (according to Ensembl, release 61 [9]) are we talking about? Which version of the sequence are referring to? Has the sequence we are annotating received any post-transcriptional modification? Genomic sequences present the same problems: although we usually talk about THE human genome, multiple versions exist, since new versions appear as the sequence is being refined. It is, thus, really important to be able to uniquely identify the sequence being annotated.

DAS response to this problem are coordinate systems. A coordinate system provides a mechanism to uniquely identify a set of sequences, so the conjunction of the coordinate system plus the sequence identifier is unique to a specific sequence. For example, there are many sequences that share 1 as its identifier. However, if we specify that it refers to a chromosome and that it's a human sequence, the number is heavily reduced. It is only necessary to say that we are referring to a sequence on the latest assembly of the human genome, GRCh37 (http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/), to uniquely define a sequence. Coordinate systems are used to ensure that features being integrated have been annotated on the same reference sequence. Although the DAS specification does not force every data source to have a coordinate system stated, it is fundamental to precisely assign one so the data on the source can be integrated with that coming from other sources. A coordinate system, however, does not have information about the identifiers of the sequences themselves. The maintenance of such list is the responsibility of the Reference Server.

A coordinate system is formally a set of properties:

Category The category the sequences on coordinate system belong to. Although the category names are not part of a controlled vocabulary, the actual names are usually part of a relatively reduced set: *Chromosome*, *Contig*, *Gene_ID*, *Protein Sequence* and a few more.

Authority The authority is the entity defining the coordinate system. Although most of the coordinate systems are defined and backed by big institutions or consortiums, like the Genome Reference Consortium or the NCBI, there are no limitations on its value. For example, a group sequencing a new organism may define a coordinate system for it and state themselves as the authority.

Species For those coordinate systems containing sequences from only one organism, the name of that organism should be stated. Thus, the coordinate system for the human chromosomes will have *Homo sapiens* as species, while it will be empty on the coordinate system containing all the proteins on UniProt.

Version While some of the underlying sequences are themselves versioned -e.g. UniProt protein sequences-, for those that are not, it is possible to specify a version. For example, the NCBI 36 human genome assembly (also known as *hg18*) will have *Authority* NCBI and *version* 36.

The DAS registry contains a complete list of all the available coordinate systems and predefined coordinate systems are available for most of the sequenced organisms.

3.3.2 Reference Servers

While the DAS network is based on the concept that all the servers are equal, some of them play a special role: the Reference Servers. As seen above, Coordinate systems are used to describe a set of sequences so they can be uniquely identified. The identifiers of the sequences themselves, however, are not specified by the coordinate system.

The Reference Server is the one in charge of providing the core information for the sequences on a specific coordinate system: their identifier and their sequence.

The identifier is the name for the sequence that will be used to uniquely identify a particular sequence in a coordinate system. It is defined by the reference server and is shared by all the annotation servers providing data for that sequence. A reference server must implement the `entry_points` command that will return a document with the sequence identifiers, the entry points to the annotations. For example, the reference server for GRCh37 is http://www.ensembl.org/das/Homo_sapiens.GRCh37.reference/ and it defines the identifier to the chromosome sequences as: *1, 2, 3, 4, . . . , 21, 22, X* and *Y*. Thus, any annotation source providing data for the chromosomes in the GCRh37 will use *1* to identify the first chromosome instead of *chr1* or *chromosome1* or any other identifier. In addition to those identifiers, if we issue an `entry_points` command to that server, we will get other identifiers such as *MT* for mitochondrial DNA and *HG183_PATCH* for one of the patches on the reference sequence.

In addition to `entry_points`, a reference server has to implement the `sequence` command. It is used to retrieve the actual sequence being annotated, usually DNA or protein sequence. For example querying the same GRCh37 reference server (http://www.ensembl.org/das/Homo_sapiens.GRCh37.reference/) for the sequence on chromosome *1* from base 1.000.000 to 1.000.020 will return the nucleotides on that range on the reference sequence (*tgggcacagcctcacccagga*) but querying the reference server for the UniProt protein sequence (http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/sequence?segment=P38398:1,20) and asking for the first 20 aminoacids of the BRCA1 protein (Uniprot identifier P38398) will return *MDLSALRVEEVQNVINAMQK*.

Although logically differentiated, Reference Servers do not need to be physically differentiated from standard Annotation Servers. Reference Servers are simply standard servers implementing the `entry_points` and `sequence` commands and with a special role on the DAS ecosystem, but nothing more than that. Usually, they also implement the `features` command and use it to state the relative position between different types of entry points, giving the smaller ones as annotations on the bigger ones. For example, a genomic Reference Server with *chromosome* and *contig* as entry point types may offer the contigs as annotations on chromosomes so their relative position can be obtained with a `features` request.

Reference Servers are already available for most of the coordinate systems and won't be usually necessary to set up one of them, however, if necessary because we are working with a new coordinate system or a newly sequenced organism, it can as simple as setting up a new source implementing the `entry_points` and `sequence` commands.

3.3.3 Features Types

Annotations returned by data sources can be classified according to its type. For example, a source annotating gene transcripts might classify its annotations in *exons*, *introns*, *utr5* and *utr3*. This information can be used to filter by type (i.e. Give only the 3'UTR in the selected region) or to change its graphical representation. It is also possible to define categories to group together related feature types. For example, the transcript annotation source could have more than one exon type (*exon:constitutive* and *exon:alternative*) but could group them together in the *exon* category and use this information to define its rendering style.

DAS fully supports the definition of types to annotate the features in an annotation server and even has a command to retrieve the types information for the features in a data source or in a region. Ontology terms are recommended when defining a type and the specific `cvId` attribute should contain a valid ontology term. Using ontology terms as types identifier allows clients to implement smarter filter capabilities. For example, a user could ask sources to retrieve features annotating non-coding genes (type *ncRNA_gene*, *SO:0001263*) and the client would retrieve features annotated with any of its descendant types (*miRNA_gene*, *piRNA_gene*, *snoRNA_gene*...).

The XML document defining the types server by a DAS source looks like this:

```
<TYPE id="SO:0001084" cvId="SO:0001084" category="inferred from electronic annotation (ECO:00000067)">2</TYPE>
<TYPE id="BS:01034" cvId="BS:01034" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
<TYPE id="BS:00138" cvId="BS:00138" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
<TYPE id="BS:01040" cvId="BS:01040" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
<TYPE id="BS:01003" cvId="BS:01003" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
```

3.3.4 Stylesheets

One of the recurrent ideas behind the DAS protocol is that data providers are the ones that know the most about their particular data. This is the reason why they are in charge of keeping data up to date and relevant, and why they define all the metadata defining it. In addition to defining the exact biological sequences they are referring to (via the assignation of a coordinate system), data providers can also define how they think the data should be represented, how should it look like when rendered on a client.

Stylesheets are special XML documents used to describe how each feature type on a source should be formatted by the client. Stylesheets can be requested by the clients with a special command, `stylesheet`. Its use is not enforced on any of the involved parts: data servers are not forced to provide a stylesheet for every source and clients are not required to honor it. The client can always override, partially or completely, the stylesheet specifications and represent the data as deemed necessary. It is important to always consider stylesheet documents as recommendations, since different clients have different attitude towards them and the actual client implementation may even not be able to strictly follow it.

Formats are defined in a per type basis with an additional one, *default*, used for those features with no type specified or whose type is not associated to any format.

Format definitions are based on a set of predefined glyphs. For each type, a glyph is selected and its parameters defined. Glyph parameters define its visual visual properties, such as color and size, and other specific properties such as orientation, presence of a label or whether they bump or not. Here, bumping refers to whether features may overlap or stack when more than one occupies the same position in the sequence.

The DAS specification does not precisely define each glyph but a general idea of what it should look like and its final exact shape is left to the implementator. Since glyphs mainly define quite standard concepts and shapes no thorough definition is needed. For example, the `ARROW` glyph can have multiple shapes, with different heads, line widths... but still be easily identifiable as an arrow.

The glyph name list include standard shapes such as as `ARROW`, `BOX`, `CROSS`, `TEXT`, `TRIANGLE`... but also includes some more specific like `GRADIENT` and `LINEPLOT`. Both `GRADIENT` and `LINEPLOT` are used to represent a series of numerical values -e.g GC content, expression levels...-. `GRADIENT` represents each value as a vertical line of a solid colour and packs all lines together, much like a heatmap would do. `LINEPLOT` represents the same data as a a line passing through the data values.

One of the limitations found on the current stylesheet specification is that it is not possible to specify different representation of the same data at different zoom levels. For example, when viewing the results from a massively parallel sequencing experiment, one could want to display the complete reads when close to base level to make visible any discrepancy or mutation present, while on wider

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="/das/das.xsl"?>
<!DOCTYPE DASSTYLE SYSTEM "http://www.biodas.org/dtd/dasstyle.dtd">
<DASSTYLE>
<STYLESHEET version="1.0">
  <CATEGORY id="component">
    <TYPE id="chromosome">
      <GLYPH>
        <HIDDEN>
        </HIDDEN>
      </GLYPH>
    </TYPE>
    <TYPE id="contig">
      <GLYPH>
        <BOX>
          <BGCOLOR>contigblue1</BGCOLOR>
          <FGCOLOR>contigblue1</FGCOLOR>
        </BOX>
      </GLYPH>
    </TYPE>
    <TYPE id="default">
      <GLYPH>
        <BOX>
          <BGCOLOR>black</BGCOLOR>
          <FGCOLOR>black</FGCOLOR>
        </BOX>
      </GLYPH>
    </TYPE>
  </CATEGORY>
</STYLESHEET>
</DASSTYLE>
```

Figure 3.6 Example of stylesheet XML document This XML has been extracted from the one served for the Ensembl reference server for the human GRCH37 assembly. Three different formats are defined for two specific types (*chromosome* and *contig*) and a special *default* one to be applied to those features without a specified type or to those whose type has no specific format.

zoom levels it could be preferable to represent them as a histogram, reducing the noise and making the overall shape more prominent.

A complete list of the available glyphs and their parameters is available on the DAS 1.6 specification.

3.3.5 Capabilities

Although the original idea of DAS was to have a number of very similar servers to be queried by the clients, who should not need to make any distinction between servers, not all servers are exactly the same. As DAS evolved and new commands and capabilities were added to the protocol, it was seen that not all servers needed to implement everything.

It is clear, for example, that while a server providing information about proteins may greatly benefit from implementing a command to retrieve structural information, that kind of information might not make sense for a genome data source or even for a protein annotation source with no information about structures.

As a consequence, the DAS specification defines a number of commands and specific characteristics for servers. Some of those are mandatory and others are optional. Each source must implement the mandatory part of the standard but is completely free to implement any subset of the optional part. To instruct the clients about what can they ask to the source and how, the server have to list exactly what part of the standard has been implemented and what is offering: its `capabilities`.

It is important to note that capabilities are used to give information not only about the commands implemented by the source or server but also about other characteristics: what error conditions will it report or whether it will accept requests from an internet browser using Cross-Origin Resource Sharing (CORS) or will require the use of a server proxy.

There are various ways of publishing a list of the available `capabilities` and usually all of them have to be available. Clients can get that information either from the HTTP special headers or the sources document or even from the DAS registry.

3.3.6 DAS commands

To retrieve different kinds of information from a DAS server different commands have to be issued. The specification of the version 1.6 of the DAS protocol defines 7 different commands. Of those, 6 are to be issued at the source level, it is, to the specific data source to be queried, while there is one special command, `sources`, that has to be issued at the server level and is used to retrieve a special document describing the sources available on that server.

A more detailed description of the commands available in the DAS specification version 1.6 can be found at the appendixIII and the complete definition in the DAS specification[16].

3.4 Ontologies

Ontologies provide semantic annotation to DAS features and are mainly used on the definition of types. While initial versions of the protocol used a set of identifiers contained on a semi-controlled vocabulary, the latest versions strongly recommend the addition of ontological terms to the definition of feature types. Ontology meta-annotation facilitates data integration between different sources and can be used to provide smart and useful filtering. In DAS, ontologies are currently used on the `id` and `cvId` attributes of a type definition (where usually the Sequence Ontology[58] and the Protein Feature Ontology[59] are used) as well as in the `method` attribute of a feature, where the ECO[60] is used to annotate how the feature was derived.

As an example of the use of ontologies, this is the response to the `types` command by UniProt when restricted to the protein BRCA2:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASTYPES SYSTEM "http://www.biodas.org/dtd/dastypes.dtd">
<DASTYPES>
  <GFF version="1.0" href="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/types?segment=Q9H265">
    <SEGMENT id="Q9H265" start="1" stop="58" version="e5f7f140a72d3f555c833992c63e910c">
      <TYPE id="SO:0001084" cvId="SO:0001084" category="inferred from electronic annotation (ECO:00000067)">2</TYPE>
      <TYPE id="BS:01034" cvId="BS:01034" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:00138" cvId="BS:00138" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:01040" cvId="BS:01040" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:01003" cvId="BS:01003" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
    </SEGMENT>
  </GFF>
</DASTYPES>
```

Chapter 4

easyDAS: Automatic creation of DAS servers

As outlined on the previous chapter, one of the lacking parts of the DAS ecosystem is easy server creation. Creating DAS servers is not as easy as initially intended and this might be discouraging groups who could be publishing and sharing their data are not doing it. Developing a system allowing these groups to publish their data without effort can potentially increase the amount of biological data publicly available to the scientific community.

We decided to take on this challenge and design and develop a software system to help scientists setting up new DAS sources with their data.

4.1 Automatic creation of DAS servers

easyDAS is a web application for the automatic creation of DAS servers. The aim was to build an easy-to-use hassle-free application allowing users with limited knowledge about DAS and network computing to set up a DAS server. Assuming the user has a data file in any of the valid formats (GFF, CSV, etc), he can upload it to the tool, define what the data represents and a new DAS source will be automatically created.

The developed system is an open source web application. An instance of it is running on the EBI systems and is completely free to use.

4.1.1 DAS server creation

Setting up a DAS server is not as easy and painless as it could be. While some DAS server implementations exist[22,61], setting up a DAS server involves quite a bit of work and requires specific know-how and facilities not available for every research group.

To set up a public DAS server, a computer visible from the network (Internet if the server is to be publicly available or simply used from a web based client such as Ensembl) is needed. Many wet-lab groups does not have those machines available since they rely on shared centrally administered systems to publish their web sites and other relevant information. Those centrally administered systems are usually closed and do not allow installation of external server programs.

For those with an available machine, installing a DAS server might require a fair amount of work. The server software needs to be installed (which usually means installing additional software packages) and configured.

All the current DAS servers are capable of connecting to multiple data backends, but this flexibility requires additional work from the user, since most of the time (specially for non-trivial implementations) the data access layer need to be implemented from scratch or at least on of the few examples need to be adapted to that particular setting. The implementation of the data access layer have to be done in the language used by the server, usually Java or Perl, and many data generating research groups does not have anyone with sufficient programming skills.

An additional database data loading step is required whenever the data is not file-based, and since file-based back-ends tend to be slow even for medium sized datasets, it is almost always the the case. If the database scheme used is not standard or does not provide specific data loading scripts, a method to load the data into the database will have to be implemented in some way.

Once everything is on place, it will have to be managed and maintained in good shape, protected from intruders, with the software up to date and other general system management work. This represents a long term commitment to the maintenance of the DAS source, specially if it is referred somewhere and so is needed to be kept alive.

This facilities, skills and resources are available for big groups and institutions, but small and medium groups or wet-lab groups, specially those without the support from a bioinformatician usually lack them. This means that nowadays many groups who could potentially publish valuable data are not doing it, at least in part because of these requirements.

One option (that is being worked on) is creating software bundles containing the DAS server, the database software and additional helper programs to load data into the database. These solutions, however, does not alleviate the need of a machine accessible from the Internet nor reliefs of its managing.

4.1.2 A hosted solution for DAS servers

easyDAS approximation to this problem is to build an externally hosted solution where groups producing biological data can create DAS sources without hitting any of the difficulties outlined above.

A hosted solution such as easyDAS frees the publisher from the need of a machine since all software will be running on the remote site. In addition, no management or maintenance work is needed from the user (apart from that directly related to data updates) as the machine itself and the instance of easyDAS will be managed by those offering the easyDAS service.

The user will not need to take care of implementing a data access layer for the DAS server, since easyDAS includes the database schema, the DAS server and the adaptor so the two of them can communicate.

easyDAS also has a web-based interface to manage the process of loading data into the database from a plain-text file. Creating that data file and defining it and its data using the web interface is the only thing required from the user.

A hosted solution, such as easyDAS, usually impose additional limitations to the user. The commands available for each data source are fixed and limited, the underlying system is fixed and cannot be changed (for example to optimize for a certain kind of data to be more efficiently served) and data loading processes are fixed to those offered by the system. However, that lack of choices means that the user will not need to take any of theses decisions and will not need any knowledge of the tools the system is based on. This effectively means than much more data providers will be proficient enough to create their own DAS servers.

4.2 easyDAS overview

easyDAS is an open-source software package developed to provide a hosted solution for the automatic creation of DAS servers. The package is freely available and can be installed by any institution, but the first fully operative instance is running at the European Bioinformatics Institute (EBI), where it can take advantage of the important computing resources available. The instance is completely open and can be used by anyone with biological data to share and publish. EBI's easyDAS can be accessed at http://www.ebi.ac.uk/panda-srv/easydas/ (temporarily unavailable). Another instance of easyDAS is available at the IMPPC at http://gattaca.imppc.org/groups/eslab/easyDAS/.

easyDAS consist of three parts: a web-based client, a server to manage the interaction with the client and user's files and a DAS server to offer a valid DAS endpoint. In addition, a database accessible by the server and the DAS server is used to store the actual data as well as the easyDAS specific data, such as user management data.

An overview of the easyDAS architecture can be seen on figure 4.1.

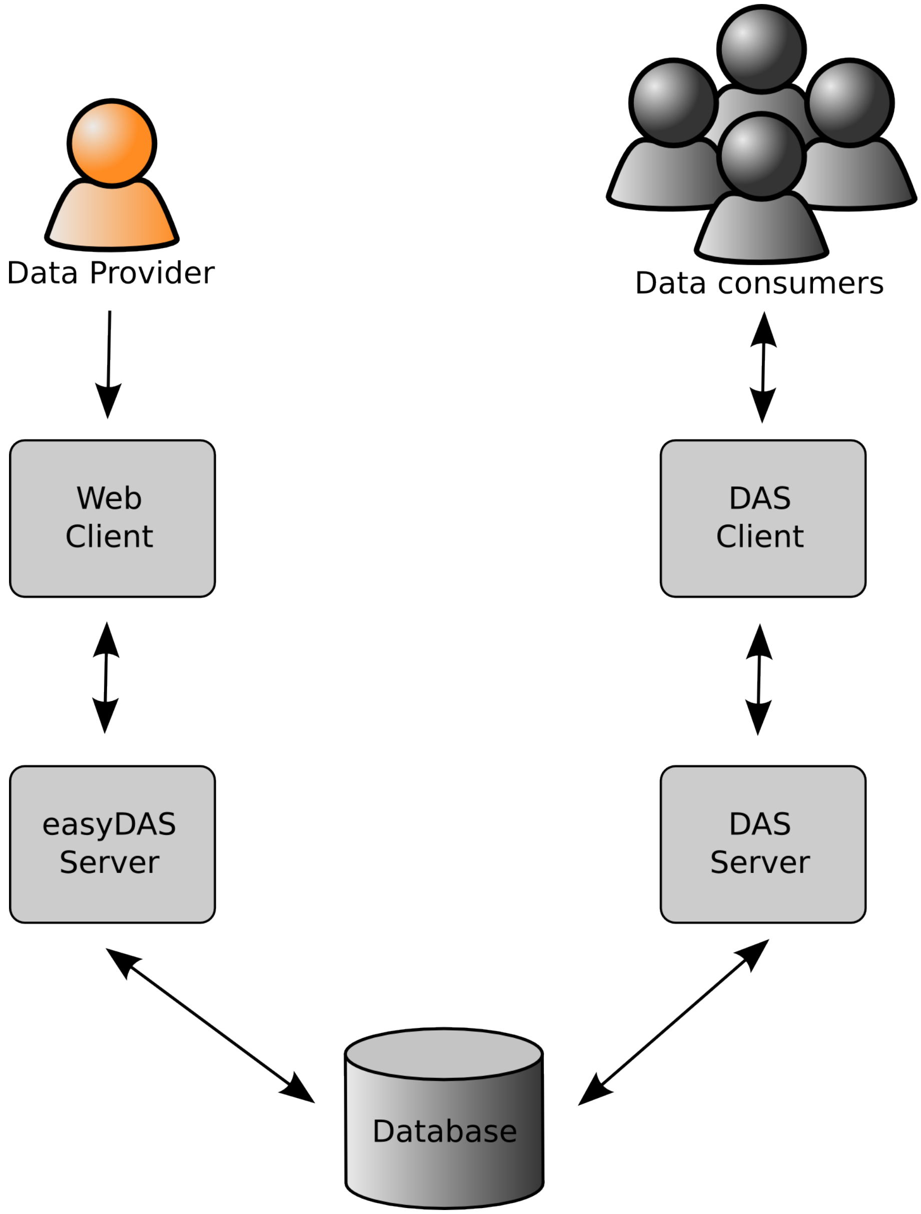


Figure 4.1 Overview of the easyDAS architecture. The basic architecture of easyDAS. The data provider interacts with the web interface to upload the data. Data is stored in the easyDAS database and automatically made available to data consumers via the easyDAS DAS server.

easyDAS has support for user registration. Although a non-registered user can create new data sources, only registered users can modify and delete them, since reliably establishing authorship is not possible. Additionally, anonymous users sources are not guaranteed to be alive for more than two weeks. Users are encouraged to register into the system to gain full access to it.

In order to created a new DAS source, the user needs a valid data file. Currently, easyDAS supports two different data files: GFF and tabulated text files (such as CSV, TSV...). The system is able to automatically identify the file type and, when dealing with a tabular file, detect the separator character used and other variants (quoted strings, etc). Those file types are easy to obtain: GFF is a format natively supported by many bioinformatics applications and creating a CSV file using any spreadsheet software such as Microsoft Excel or OpenOffice Calc is very straightforward.

Once the user has a valid data file, creating a new DAS source is usually a matter of a few clicks. easyDAS has a wizard interface to guide the user through that process step by step: data file uploading, file type validation, source description, linking to ontology terms...

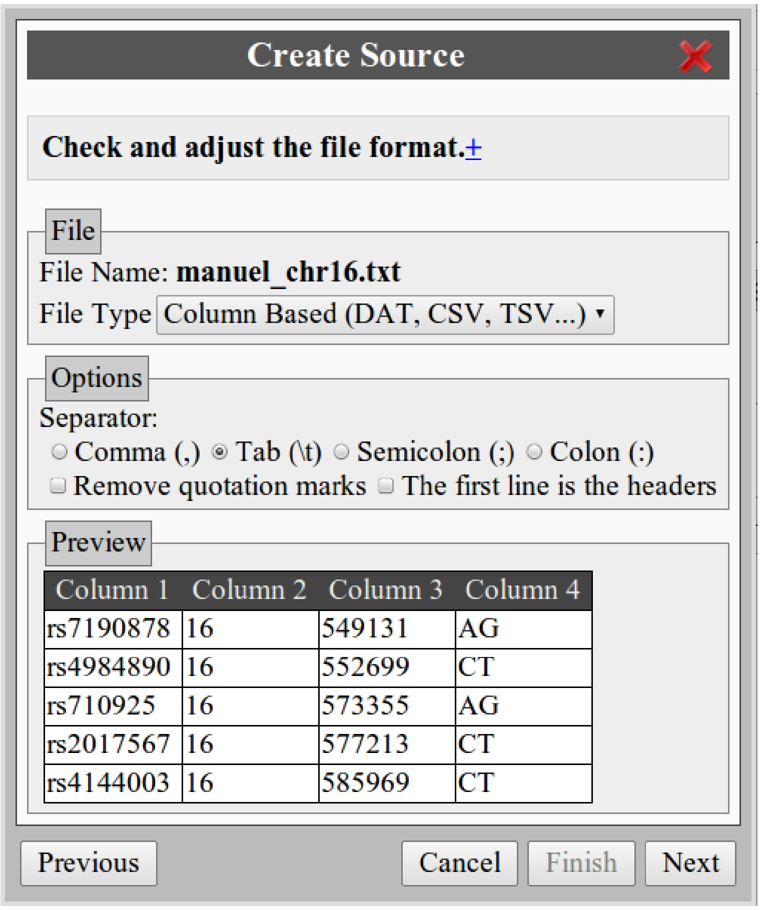
A crucial step is the exact description of the data file structure and its linking to DAS data structures (i.e. which column represents the feature start and stop, which one should be used as a label...). GFF files have a fixed field structure (except for the comments column) and so this mapping is predefined (although it might be changed by the user). Tabular files, however, do not have any kind of predefined structure and the mapping has to be user-defined. A set of heuristics are in place to prepopulate the mapping with sensible values according to the data, but it's the user who has to check and accept it.

easyDAS is free open source software under the GNU Lesser General Public License (LGPL) and the project is hosted on Google Code at http://code.google.com/p/easydas. The first instance is available at the EBI at http://www.ebi.ac.uk/panda-srv/easydas.

4.3 Features and characteristics

easyDAS was designed to be easy to use. The main part of the interface is a simple list of the available sources annotated with their metadata -description, coordinate system, maintainer email- along with a link to the DAS source itself and, for sources annotating the most commonly used organisms, a direct link to the ensembl genome browser with the DAS source already attached to facilitate the exploitation of the data. In addition to the main sources table only three buttons are available: `Login`, `Help` and `Create a new source`.

The main use case is triggered by the `Create a new source` button. It is structured as a standard wizard interface guiding the user though a series of simple steps: upload a data file, tweak and validate the parsing mode, defining the source



File Name: **manuel_chr16.txt**

File Type: Column Based (DAT, CSV, TSV...) ▾

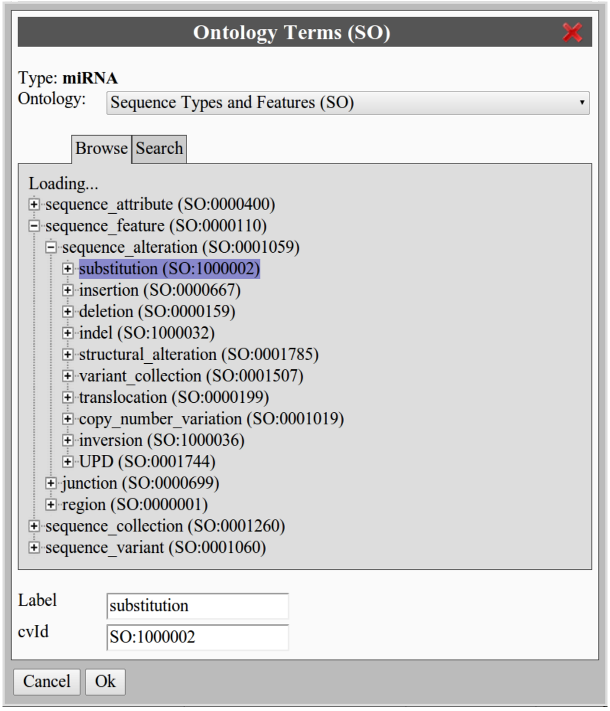
Separator:
○ Comma (,) ● Tab (\t) ○ Semicolon (;) ○ Colon (:)
□ Remove quotation marks □ The first line is the headers

Column 1	Column 2	Column 3	Column 4
rs7190878	16	549131	AG
rs4984890	16	552699	CT
rs710925	16	573355	AG
rs2017567	16	577213	CT
rs4144003	16	585969	CT

Figure 4.2 easyDAS source creation wizard. To create a new source, a wizard-like interface is available.

and selecting a coordinates system, defining the mapping between the data in the file and feature attributes used in DAS and optionally assigning default values for any unspecified value and selecting the ontology terms associated to attributes accepting an ontology term -i.e. feature types and methods-. The interaction is done via a series of forms in floating dialogs (Figure 4.2 with inline filling instructions and help popups and a classical **“Next”** and **“Previous”** course of action.

In addition to the standard wizard guided path, two additional dialogs have been created, one to select the coordinates system that includes a list of all available DAS coordinates systems and means to filter them by organism, authority and source. The second additional dialog is a custom made ontology browser based on the Ontology Lookup Service (OLS)[21] (Figure 4.3). The ontology browser


Ontology Terms (SO)
Browse
Search
Loading...
sequence_attribute (SO:0000400)
sequence_feature (SO:0000110)
sequence_alteration (SO:0001059)
substitution (SO:1000002)
insertion (SO:0000667)
deletion (SO:0000159)
indel (SO:1000032)
structural_alteration (SO:0001785)
variant_collection (SO:0001507)
translocation (SO:0000199)
copy_number_variation (SO:0001019)
inversion (SO:1000036)
UPD (SO:0001744)
junction (SO:0000699)
region (SO:0000001)
sequence_collection (SO:0001260)
sequence_variant (SO:0001060)
Cancel
Ok

Type: **miRNA**
Ontology: Sequence Types and Features (SO)

Label: substitution
cvId: SO:1000002

Figure 4.3 Browse Ontology Dialog. To encourage the annotation of data with ontology terms, easyDAS offers an ontology borwser dialog based on the Ontology Lookup Service (OLS).

allows one to select any of the ontologies available in the OLS and has both a tree browser to interactively navigate the ontology terms and a search option to retrieve all terms matching the specified criteria.

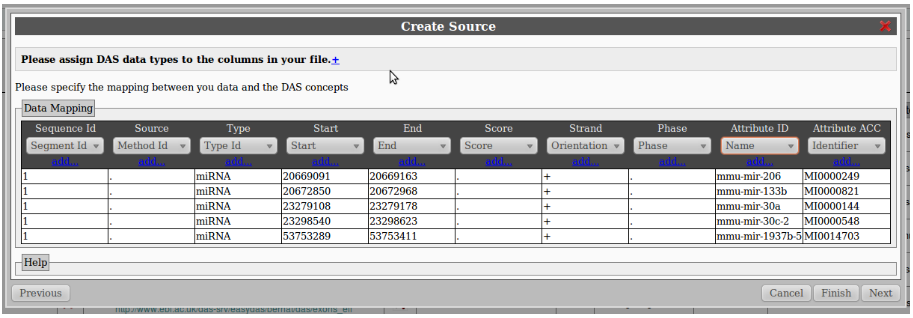
At this time, easyDAS has support for two different file formats: the General Feature Format (GFF) and tabular formatted plain text files including CSV and tab delimited formats. Adding support for new formats i pretty easy and only requires extending some of the back-end modules to instruct them on how to parse the new file format and creating one HTML form to populate the format specific wizard step. For the GFF format, both GFF version 2 and GFF version 3 files are supported and automatically detected and the *attributes* field is automatically detected and expanded into multiple fields even for most non standard but widely used syntaxes. In addition, a set of custom header comments are available. Custom header comments are read in and interpreted by easyDAS and used to prepopulate some of the data fields needed when creating a new source. It is possible, for example, to include a `##source-name` line to specify the source name, a `##source-title` to prepopulate the title of the source or a `##source-maintainer`

```
##gff-version 2
##source-name mmu_mirna
##source-title Mmu microRNAs
##source-mantainer bernatgel@gmail.com
#
# Chromosomal coordinates of Mus musculus microRNAs
# miRNA data:       miRBase Sequence (version 16)
# Genome assembly:  NCBIM37
#
1       .       miRNA       20669091       20669163       .       +       .       ACC="MI0000249"; ID="mmu-mir-206";
1       .       miRNA       20672850       20672968       .       +       .       ACC="MI0000821"; ID="mmu-mir-133b";
1       .       miRNA       23279108       23279178       .       +       .       ACC="MI0000144"; ID="mmu-mir-30a";
```

Figure 4.4 GFF file example. An example of a valid GFF file that would be automatically recognized and configured by easyDAS. In addition to the standard header, this GFF includes easyDAS specific header comments used to prepopulate the source creation wizard.

with the maintainer email. This is a useful feature when data files are the result of an automated pipeline and human intervention when creating sources should be minimized. Figure 4.4 shows an example of the header and the first few lines of a GFF data file with easyDAS custom headers. The second supported format are tabular files. easyDAS tests and automatically detects a set of standard field delimiters -tabs, spaces, commas, semicolons...- but it is possible to specify any character as the delimiter in the file format parameters dialog. Quotation marks and column headers are also detected automatically and set in the file format parameters dialog.

A key step in the source creation wizard is that of mapping. DAS has specific set of attributes per feature -*id*, *method*, *type*, *start* and *end*, *score*- and some of them, such as *note*, might have multiple values per feature. In order to give more flexibility to the user when creating the data files an interface to specify the mapping between the fields in the data file and the DAS feature attributes is available. Whenever it is possible, a mapping is suggested so the user only needs to check the validity of the mapping and click *"Next"*. These suggestions depend partially on the file format -e.g. GFF fields have a natural mapping into DAS attributes- and on a set of heuristics computed over the field headers that, although quite simple, give in general quite good mapping suggestions. The available interface allows the modification of the mappings, assigning more than one field to the multiple value accepting DAS attributes and leaving DAS attributes blank. A screenshot of the mapping step can be seen on figure 4.5. An additional feature of easyDAS giving more flexibility to users when creating data files is the defaults system. With it, is it possible to specify default values for any DAS attribute and specify whether the default should be applied to all features in the file or only to those with no value. Thanks to the defaults system it is possible to create simpler and smaller data files with only the relevant information. For example, if we had a set of single nucleotide variations of the same type and found using the same methods, we could create a data file with only two columns, chromosome and position, assign



Create Source

Please assign DAS data types to the columns in your file.+

Please specify the mapping between you data and the DAS concepts

Data Mapping

Sequence Id	Source	Type	Start	End	Score	Strand	Phase	Attribute ID	Attribute ACC
Segment Id	Method Id	Type Id	Start	End	Score	Orientation	Phase	Name	Identifier
add...	add...	add...	add...	add...	add...	add...	add...	add...	add...
1	.	miRNA	20669091	20669163	.	+	.	mmu-mir-206	MI0000249
1	.	miRNA	20672850	20672968	.	+	.	mmu-mir-133b	MI0000821
1	.	miRNA	23279108	23279178	.	+	.	mmu-mir-30a	MI0000144
1	.	miRNA	23298540	23298623	.	+	.	mmu-mir-30c-2	MI0000548
1	.	miRNA	53753289	53753411	.	+	.	mmu-mir-1937b-5	MI0014703

Figure 4.5 The mapping window. This part of the DAS source creation wizard of easyDAS consists of a dialog asking the user to map the uploaded data fields onto DAS concepts. A set of heuristics is in place to automaically propose possible mappings to the user.

the position to *start* and *end* in the mapping dialog and define default values for *method*, *type*, etc.

With easyDAS it is possible to create two different types of sources: public and private. Public sources are open, available to anyone and listed in the easyDAS main page. Private sources are exactly like the public ones -open and available- but are not listed in the main page, so one needs to know the source url to gain access to it. It is important to emphasize that there is no source security implemented and both private and public sources are freely available to anyone, the only difference between them being the listing in the main page source list. Only registered users can create private sources and all sources created by non-registered users are assigned to *anonymous*. Source deletion is also restricted to registered users. Users can register either using a traditional user name and password pair -that will be securely stored in the easyDAS database- or register and log-in using an OpenID provider and in such case, easyDAS will never get access to your password and it will not be stored in the system.

In easyDAS, each registered user has it's own DAS server. When a user registers, a new DAS server is created and all new sources created by the user are located under that server. This has some interesting advantages, such as having all sources created by a user logically tied together and accessible by a standard sources command and avoiding source name clashes between sources from different users, since the uniqueness of the source name is only required inside a server.

Help and documentation is available both as separate documents and inline. The *Help* button links to the external documentation that includes a quick start guide and a tutorial. Inline documentation is available in the source creation wizard, with a general explanation of each step and a small popup description for every form field.

4.4 Implementation

easyDAS is a client-server application with two different entry points: a web-based interface to upload data and create and manage sources and a DAS server accepting DAS requests (Figure 4.1). All source code is under the LGPL open source license and can be found at http://code.google.com/p/easydas.

The most complex part of the system is the server. It has been implemented with perl as a modular system, with a central controller, easyDAS.pl, managing the interaction with the web interface and coordinating the rest of the modules. The key process performed by easyDAS is the creation of a new DAS server and this is made in a coordinated manner by the client and the server. This process includes the heuristic determination of the file format and its parameters, the available fields for each element, a tentative mapping to DAS elements derived from the column headers and the mapping to ontology terms. After all metadata about the new data source have been compiled, a new source is created in the database and all data is moved into the database.

The database used is a MySQL -although all communication with it is done via the DBI module, so any other supported database could be used-. The number of features per source is expected to range from a handful to potentially millions and different sources are completely independent. In addition, it is expected that requests for data from a specific data source will come in bursts, with long periods of no activity and short periods with lots of requests corresponding to a browsing session by a data consumer. Thus, to prevent big sources from affecting small ones and to improve cacheability, a set of tables is created for every source -features, types, segments, etc...-. With this structure, data from each source is stored independently and data requests hit only the source specific tables.

The web client has been implemented as a single page application using jQuery[23] as a base library. It is modular and event-based, with a controller class managing the coordination between different modules and the communication with the easyDAS server. The interface elements and widgets are custom-made and not based on any library. A custom ontology browser based on the OLS[21] web-services interface has been written. easyDAS offers two different methods for user authentication: classic user and password and OpenID. With OpenID it is possible to sign in to easyDAS using any openID identity provider, so it is possible to register using, for example, a Google Account. OpenID support has been implemented using the `Net::OpenID` package.

The last part, the DAS server, is a customized version of ProServer. ProServer is a DAS server written in perl and is the server behind the Ensembl DAS sources. The normal way in ProServer of defining DAS sources is via modifications of a configuration file, which requires a server restart for them to take effect. However, it also has a useful functionality, called *hydra*, that allows it to serve multiple sources based on a single entry in the configuration file and retrieving the specific source information by other means, for example, from a database. From a DAS point of view, in easyDAS each user has his/her own complete and independent DAS server, preventing any source name clash and grouping all sources from a user in a single `sources` response without interference from other users data. To achieve this the *hydra* infrastructure of ProServer was modified to simulate independent servers instead of sources.

4.5 Expected usage

The envisioned usage of easyDAS encompasses three different users groups.

The first and foremost user type easyDAS should cater are those researchers producing valuable data but unable to share it via DAS due to the problems derived of setting up a DAS server. The main goal when designing easyDAS has been making the system usable by this type of not necessarily computer savvy users in order to convert them from data producers into data sharers.

Another expected usage of easyDAS would be to create a DAS server as a companion to a publication. For example, a classical microarray experiment usually produces a list of differentially expressed genes or an aCGH would produce a list of regions with an altered copy number. Creating a DAS server with that data and referencing it on the publication would make those datasets easily available to anyone and free the research team of any long term data maintenance. Sharing research results on a computer friendly way could increase its visibility and real value, helping other groups on working upon them.

Finally, creating a DAS source can be an easy way of sharing annotations with other team members and integrating them into third party resources. For example, once a DAS source has been created, integrating it into Ensembl browser is as easy as clicking on the Ensembl logo on the sources list. Sharing such a source might be a practical way for bioinformatics units to return experiment results to their users, in addition to the classical data files.

In addition, it is possible for a user to set up his own easyDAS instance. easyDAS is open source and freely available at http://code.google.com/p/easydas. So far, there are no installation instructions and setting everything up correctly can require a bit of work. We expect to create an initial installation guide in the short future. easyDAS can be a good addition to the services offered by many research centers to their researchers as a way to use and visualise their data in external tools such as web browsers or other analysis services accepting DAS data as output. The creation of a DAS source by an institution and filling it with interesting data generated by their researchers might also increase its visibility and help in the dissemination of the work performed there.

4.6 Example Usage

To illustrate the usage of easyDAS we will present an example. In it, we will see the creation of two DAS sources from two data files. These sources will be later used to show a usage example with GenExp, and to describe a possible path to discovery.

4.6.1 Data Files

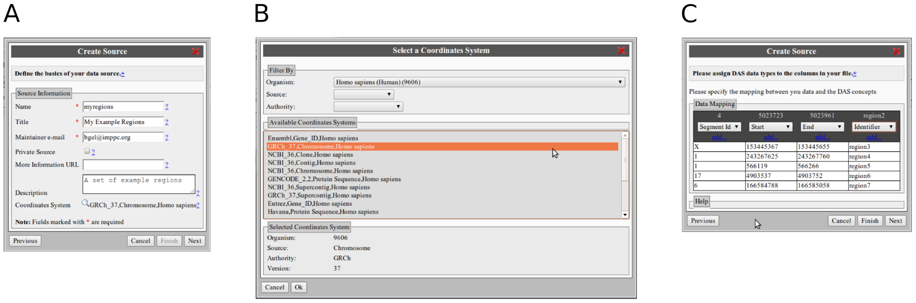
One data file representing data generated by a researcher is used in this example.

ChIP.Peaks.txt This file contains a set of regions representing the peaks called by a peak calling algorithm applied to the data generated by a ChIP-seq experiment. The contents of this file have been generated and do not correspond to a read experiment. The file is a tab separated file with four columns (chromosome, start, end and name) and contains 478 records. This is an example of the first few lines of the file.

```
chr        start        end        name
1        11162714        11162862        region1
4        5023723        5023961        region2
X        153445367        153445655        region3
1        243267625        243267760        region4
1        566119        566266        region5
17        4903537        4903752        region6
```

4.6.2 Background

This example represents the workflow of scientist who performed a ChIP-seq experiment to study the distribution of his favorite DNA-binding protein on the genome of a certain cell line. After the primary bioinformatics analysis -quality controls, read preprocessing, mapping and finally peak calling- he has a BED file with the positions of the peaks in the genome. His aim is to be able to visualise the positions of these regions along the genome and to study them and their relation with other genomic features. To do that, he will use different programs and among them, genome browsers to visualise his regions. Since many genome browsers accept DAS sources as a way to add custom data, he will start creating a new DAS source with easyDAS.



Create Source

Define the basics of your data source.+

Source Information

Name * myregions ?

Title * My Example Regions ?

Maintainer e-mail * bgel@imppc.org ?

Private Source □ ?

More Information URL ?

Description: A set of example regions ?

Coordinates System GRCh_37,Chromosome,Homo sapiens?

Note: Fields marked with * are required

Select a Coordinates System

Filter By

Organism: Homo sapiens (Human) (9606)

Source:

Authority:

Available Coordinates Systems

Ensembl,Gene_ID,Homo sapiens
GRCh_37,Chromosome,Homo sapiens
NCBI_36,Clone,Homo sapiens
NCBI_36,Contig,Homo sapiens
NCBI_36,Chromosome,Homo sapiens
GENCODE_2.2,Protein Sequence,Homo sapiens
NCBI_36,Supercontig,Homo sapiens
GRCh_37,Supercontig,Homo sapiens
Entrez,Gene_ID,Homo sapiens
Havana,Protein Sequence,Homo sapiens

Selected Coordinates System

Organism: 9606
Source: Chromosome
Authority: GRCh
Version: 37

Create Source

Please assign DAS data types to the columns in your file.+

Please specify the mapping between you data and the DAS concepts

Data Mapping

4	5023723	5023961	region2
Segment Id	Start	End	Identifier
X	153445367	153445655	region3
1	243267625	243267760	region4
1	566119	566266	region5
17	4903537	4903752	region6
6	166584788	166585058	region7

Help

Figure 4.6 Screenshots of the different dialogs used in the source creation process. A: The source metadata dialog. B: Coordinate system selector. C: Mapping window.

4.6.3 Example

To create a DAS source with easyDAS connect to an easyDAS instance, for example the one at http://gattaca.imppc.org/groups/eslab/easyDAS/, and login or register so your sources are not anonymous. Once done, click the *"Create a new source"* button, and the *Create Source* dialog will appear. Select your data file, in our case, `ChIP.Peaks.txt` and click on upload. The file format will be detected automatically. Click on next and change the format options if needed. In the next step, you will need to add the source metadata (Figure 4.6 **A**) such as name, description maintainer and select a coordinate system (Figure 4.6 **B**), in this case GRCh37 since it's human data. After that, he will have to map the columns of his data file into the different DAS concepts (Figure 4.6 **C**). The next windows will help assigning any default value not present in the original data file and using an ontology browser, assigning an ontology term to the different feature types present in the data file. After clicking on *"Finish"*, a new DAS source will be created and the data will be ready to be explored. The exploration process will be presented in chapter 6.

4.7 Key Contributions of easyDAS

This is a summary of the key contributions of easyDAS:

Automatic DAS server creation: easyDAS is the first and only tool for the automatic creation of DAS sources. From a simple and standard data file and following a simple wizard interface a user can create a new DAS sources with just a few mouse clicks.

Almost no DAS knowledge required: It is possible to use easyDAS to create DAS sources without any knowledge of the technical side of DAS. Knowing the different attributes available and a few key concepts such as coordinates systems and the like would be helpful but not required.

Centralized computing: Takes advantage of centralized computing facilities to free data providers from the burden of maintaining a DAS server.

Web based: easyDAS is web based and the user does not need to install any additional software in order to use it. It also means it is OS agnostic and can be used from any operating system

Simple input formats: The input files needed are in standard and simple formats -GFF and tabular files- that can be produced by many bioinformatics software systems or even by a spreadsheet program such as Microsoft Excel or LibreOffice Calc.

4.8 easyDAS publication

The paper presenting easyDAS was published on BMC Bioinformatics on January 2011. It received the *Highly Accessed* mark to reflect its high number of accessions right after publication.

This work has also been presented in various congresses and workshops either as a talk (*12th Annual Bioinformatics Open Source Conference (BOSC 2011) - 19th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB 2011)* (Vienna, Austria, 2011), *DAS Workshop 2011* (Hinxton, United Kingdom, 2011) and *DAS Workshop 2010* (Hinxton, United Kingdom, 2010)), a demonstration (*International Symposium on Integrative Bioinformatics* (Cambridge, United Kingdom, 2010)) or as a poster (*19th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB 2011)* (Vienna, Austria, 2011)).

Author contributions: The original idea is from AMJ, RCJ and HH. Under the guidance from XM and with the helpful input from the two DAS experts AMJ and RCJ, I developed the idea and designed the system. The implementation and testing was done by me. I wrote the bulk of the manuscript, which was revised and approved by all the authors.

The actual paper follows.

Gel Moreno *et al. BMC Bioinformatics* 2011, **12**:23
http://www.biomedcentral.com/1471-2105/12/23

BMC Bioinformatics

SOFTWARE Open Access

easyDAS: Automatic creation of DAS servers

Bernat Gel Moreno[1,2*], Andrew M Jenkinson[2], Rafael C Jimenez[2], Xavier Messeguer Peypoch[1], Henning Hermjakob[2]

Abstract

Background: The Distributed Annotation System (DAS) has proven to be a successful way to publish and share biological data. Although there are more than 750 active registered servers from around 50 organizations, setting up a DAS server comprises a fair amount of work, making it difficult for many research groups to share their biological annotations. Given the clear advantage that the generalized sharing of relevant biological data is for the research community it would be desirable to facilitate the sharing process.

Results: Here we present easyDAS, a web-based system enabling anyone to publish biological annotations with just some clicks. The system, available at http://www.ebi.ac.uk/panda-srv/easydas is capable of reading different standard data file formats, process the data and create a new publicly available DAS source in a completely automated way. The created sources are hosted on the EBI systems and can take advantage of its high storage capacity and network connection, freeing the data provider from any network management work. easyDAS is an open source project under the GNU LGPL license.

Conclusions: easyDAS is an automated DAS source creation system which can help many researchers in sharing their biological data, potentially increasing the amount of relevant biological data available to the scientific community.

Background

In recent years the amount of biological data generated have been increasing greatly, and with the advent of new technologies like next generation sequencing this trend is likely to increase. Additionally, new analysis and re-analysis techniques help to produce better and more accurate derived results every day. Making all this data and results publicly available can be of great benefit for the scientific community as a whole, since valid biological data can be used both by field researchers and by those developing new methodologies and algorithms. Sharing raw research data allows others to conduct re-analysis and meta-analysis, usually reinforcing the previous results or even producing novel ones. ArrayExpress [1] and GEO [2] have had a very positive effect on microarray research development and have been heavily used in the development of both new data analysis techniques and biological knowledge. Sharing research results eases rapid spreading of new findings and its incorporation in ongoing research increasing its overall usefulness.

The effects of this sharing can be greatly increased if data and results are made publicly available using some kind of machine readable standard format allowing them to be seamlessly used by other researchers.

While making the raw data available as supplementary material attached to the publication of a paper can be useful and other researchers can certainly use it, its integration with data from other sources will still be difficult and will not help fully automatic approaches such as workflows. However, if this same data is made available in a standard machine readable format it can be easily integrated with data coming from other standard sources and automatically displayed and analyzed.

The Distributed Annotation System (DAS) is a complete system for sharing annotations on biological sequences. It comprises a standard XML based file format, an accurate definition of the semantics of the data -based on the use of ontologies of biological terms-, and an HTTP based REST style protocol for sharing those annotations [3-5]. Figure 1 is an overview of the DAS system. Many Annotation Servers can provide annotations

* Correspondence: bgel@lsi.upc.edu
[1]Software Department, UPC-BarcelonaTech, Barcelona, Spain
Full list of author information is available at the end of the article

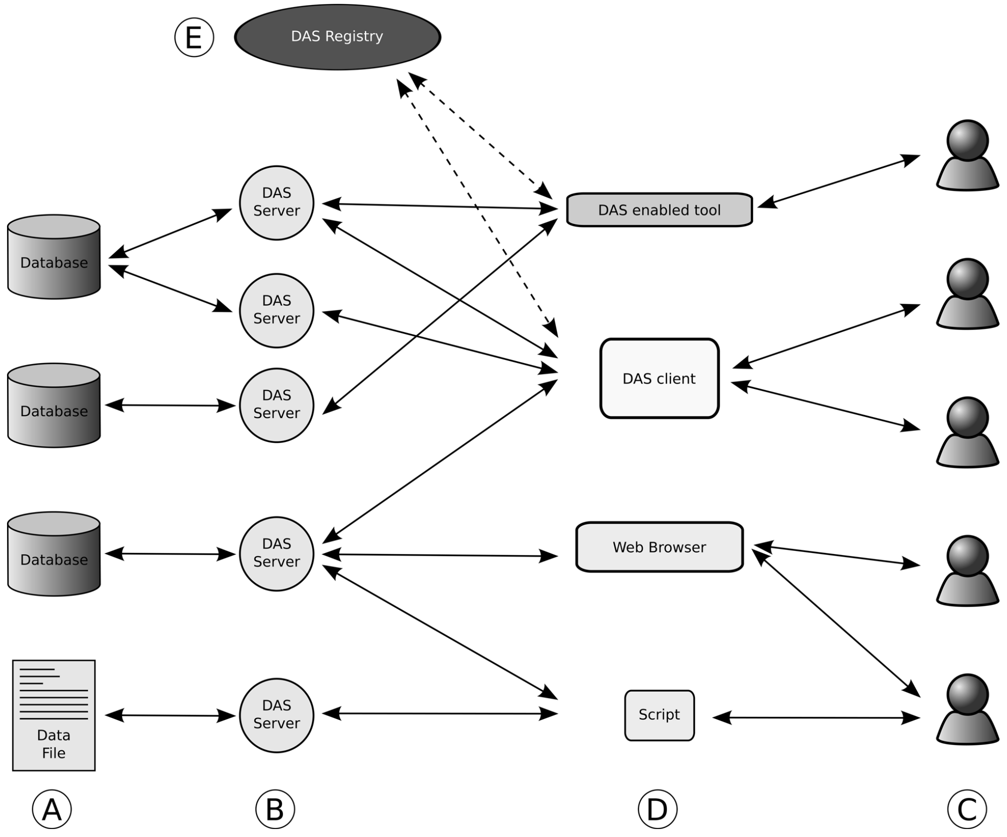


Figure 1 Overview of DAS. Data is stored in either databases or data files (A). DAS servers (B) offer a common interface to access that data, the DAS protocol. Users (C), can use different client types (D) including specific DAS clients, internet browsers and non-visual scripts, to access the data. Optionally, clients can access the DAS Registry (E) to retrieve a list of available DAS sources.

of sequence objects provided by a Reference Server, for example Ensembl or UniProt. While initially designed to annotate genomic sequences, DAS has support for both genomic and protein annotations, for sequence alignment data, and for structural information. Other federated systems exist for other types of biological data, such as PSICQUIC [6] for molecular interaction data.

DAS client-server architecture was designed around the idea of having a small number of complex clients integrating data coming from, potentially, many of different simple sources. Some examples of DAS clients are Ensembl [7], Dasty2 [8], GBrowse [9], Jalview [10], SPICE [11], PeppeR [12], DASher [13]. Sharing biological data on DAS allows data providers to leverage the DAS ecosystem and make it easy to integrate their data with other existing sources.

DAS server software is available in different programming languages, such as ProServer [14] in Perl and Dazzle [15] and MyDAS [16] in Java. However, despite the idea of DAS servers being simple, setting up a DAS server is not a trivial task. DAS servers allow for a great flexibility on where the actual data is stored and how it is structured. Usually the backend is database, but files and other options are also viable. The downside of this flexibility is that very often data providers will need to implement a custom made data access layer mapping their real data layout to the DAS concepts used in the server and this will have to be done either in Perl or Java. There are many research groups who will not have easy access to people proficient enough in programming to implement that access layer. In addition, setting up and managing an internet accessible machine to host the server can be also difficult or a too big overhead for many data generators, mainly for those with small data sets.

Thus, the challenge: converting all those data generators into data providers, increasing the amount and variety of the biological data available to the scientific community and contributing to the collective annotation of biological sequences.

Results and Discussion

easyDAS has been developed to help on that conversion offering a web-based and ready-to-use system for biological data sharing using DAS. The user only needs to upload a text file with the data into easyDAS and set a few configuration options, mainly stating what the data represents, and the system will automatically create a new DAS source. Although the DAS source will be automatically created and managed by easyDAS, the user will retain full control over the data and will be able to modify or delete it at any point using the same web interface.

We envision easyDAS being useful for a wide range of potential DAS users. Biologists producing annotations can create DAS sources to take advantage of the DAS infrastructure to integrate their data into Ensembl or other DAS clients. Creating a DAS source can also be used to spread new experimental results and share them with other researchers. Computational biologists and bioinformaticians developing new analysis or prediction tools can easily create new sources with the results of applying those new tools to known datasets making their example runs publicly available and usable. Finally, and since there is no limit on the number of sources to be created by a user, personal datasets can be also published in easyDAS, creating as many temporary sources as needed. This would allow bioinformatics service groups to create tailored sources containing, for example, the set of genes resulting from a microarray analysis or study-specific annotations of proteins. In any case, the user is relieved from the burden of setting up and maintaining their own DAS servers.

easyDAS is fully compliant with the recently approved DAS 1.6 specification [5] and encourages the new systematic semantic annotation of features via an integrated ontology browser based on the Ontology Lookup Service [17].

An easyDAS is freely available at http://www.ebi.ac.uk/panda-srv/easydas.

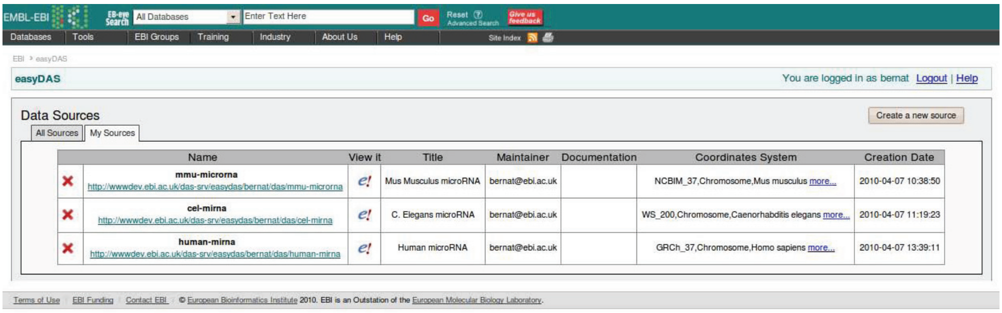
easyDAS is open source software under the GNU LGPL license. The project is hosted at Google Code and can be freely accessed and downloaded at http://code.google.com/p/easydas/.

System Description

easyDAS exposes two different entry points to the users. The first one is a web interface where sources can be found, created and managed. The other one is the DAS server, which exposes the data using the DAS protocol. The main components of easyDAS web interface are a users management system, a list of the existing data sources and a wizard for creating new sources.

The main page in the easyDAS web interface is the list of the available sources with their descriptions and URLs (Figure 2). This list offers easy access to data sources created by any user, facilitating the use and spreading of that data. Links to popular DAS clients like Ensembl are provided too, and will usually attach the source to the viewer automatically. All easyDAS sources can also be registered in the DAS Registry [18] to further increase their visibility.

To create a new DAS source based on a custom dataset, a wizard-like interface will guide the user through the whole process: uploading the file and providing some information about the structure and type of data



	Name	View it	Title	Maintainer	Documentation	Coordinates System	Creation Date
✗	mmu-microrna http://wwwdev.ebi.ac.uk/das-srv/easydas/bernat/das/mmu-microrna	e!	Mus Musculus microRNA	bernat@ebi.ac.uk		NCBIM_37,Chromosome,Mus musculus more...	2010-04-07 10:38:50
✗	cel-mirna http://wwwdev.ebi.ac.uk/das-srv/easydas/bernat/das/cel-mirna	e!	C. Elegans microRNA	bernat@ebi.ac.uk		WS_200,Chromosome,Caenorhabditis elegans more...	2010-04-07 11:19:23
✗	human-mirna http://wwwdev.ebi.ac.uk/das-srv/easydas/bernat/das/human-mirna	e!	Human microRNA	bernat@ebi.ac.uk		GRCh_37,Chromosome,Homo sapiens more...	2010-04-07 13:39:11

Figure 2 Sources list. Screenshot of the main page of easyDAS with the user logged in. The table contains a list of the sources created by the user and offers means to explore and remove them.

in the file. This wizard can be invoked clicking on the **"Create a new source"** button in the main page. Although DAS supports alignment and 3 D structure annotations in addition to the standard sequence annotations, those capabilities are currently not heavily used -about 1% of the registered sources offer them- and usually provided by specialized sources. Thus, and since the vast majority of users are willing to share mainly sequence annotations, easyDAS does not support them.

easyDAS provides a rich user interface with standard elements like dialogs and wizards resembling those of a desktop application. It is completely AJAX driven to increase usability and interactivity and to improve system response time.

User Registration

Users can register to the system using either OpenID [19] or a traditional username/password pair. All sources created by registered users can be updated, modified or even deleted after creation and can exist for an unlimited time. Sources created by unregistered users will be considered "anonymous" sources and will be online for a limited time only. Anonymous sources cannot be modified or deleted.

When a user is logged into the system, a second tab will be activated in the main page listing only the sources created by the user. The source modification and deletion functionality can be accessed from that table.

Source Privacy

It is very important to note that, as almost any other DAS source, easyDAS sources are fully public and accessible by anyone through a simple URL. Although it's possible to mark a source so it's not published in the main easyDAS list, this will not control the access to the actual data. It is not currently possible to create private DAS sources using easyDAS.

File Formats

The currently supported file formats are the General Feature Format (GFF) and a versatile implementation of the Comma Separated Values (CSV) where the separator can be any character (although only comma, semicolon and tabulator will be automatically detected). While GFF is a commonly used format that many bioinformatics software can export, CSV can be obtained from any tabular data using a spreadsheet program like Microsoft Excel or OpenOffice Calc.

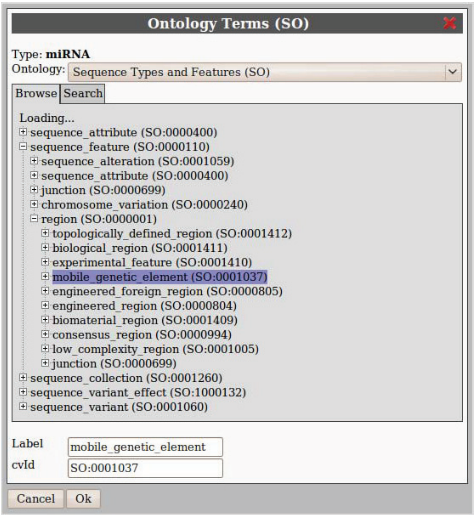
Some minor extensions and modifications to the formats are supported, like comments on CSV files.

Additionally, easyDAS specific information can be added in the form of special comments containing metadata. Those special comments can be used to pre-configure the source metadata so even less work is required on the web interface. Examples of such comments are the GFF options `source-name`, `source-title` or `source-maintainer` that will be used to pre-populate the corresponding source metadata fields.

Semantic Annotation

One of the important additions of the version 1.6 of the DAS specification is the standardization of the semantic annotation of features. Each annotation in DAS has a property describing its type (type), and another describing how it was generated (method). Both properties could already be expressed using ontology terms, but the inclusion of the cvId attribute to specify the controlled vocabulary identifier associated to types and methods and the recommendation of using specific ontologies standardizes those options and makes semantic management of DAS sources and features easier. Specifying ontology terms for feature types will improve data integration and organization in semantic-aware clients like Dasty2 that offer ontology based filtering and grouping. By having most of the sources semantically annotated, easyDAS will be well placed to take advantage of a future semantic search function in the DAS Registry.

easyDAS encourages the correct use of those new attributes and has a simple ontology browsing and searching widget (Figure 3). That widget offers interactive browsing of ontologies using on-demand leaf expansion and complete text search capabilities. All that functionality is backed by the web services provided by



Ontology Terms (SO)

Type: **miRNA**
Ontology: Sequence Types and Features (SO)

Browse | Search

Loading...
⊞ sequence_attribute (SO:0000400)
⊟ sequence_feature (SO:0000110)
  ⊞ sequence_alteration (SO:0001059)
  ⊞ sequence_attribute (SO:0000400)
  ⊞ junction (SO:0000699)
  ⊞ chromosome_variation (SO:0000240)
  ⊟ region (SO:0000001)
    ⊞ topologically_defined_region (SO:0001412)
    ⊞ biological_region (SO:0001411)
    ⊞ experimental_feature (SO:0001410)
    ⊞ mobile_genetic_element (SO:0001037)
    ⊞ engineered_foreign_region (SO:0000805)
    ⊞ engineered_region (SO:0000804)
    ⊞ biomaterial_region (SO:0001409)
    ⊞ consensus_region (SO:0000994)
    ⊞ low_complexity_region (SO:0001005)
    ⊞ junction (SO:0000699)
⊞ sequence_collection (SO:0001260)
⊞ sequence_variant_effect (SO:1000132)
⊞ sequence_variant (SO:0001060)

Label: mobile_genetic_element
cvId: SO:0001037

Figure 3 Ontology browser. The easyDAS ontology browser widget, used to select ontology terms to semantically annotate the features. It leverages the Ontology Lookup Service webservices to offer both ontology tree browsing and search capabilities.

the Ontology Lookup Service. The ontologies recommended in the DAS specification, Sequence Ontology [20], Biosapiens Ontology [21] and PSI-MOD [22] and Evidence Code Ontology, can be used to annotate the features. Any other ontology available on OLS can be added upon request.

To help in the process of selecting the right ontology terms and to improve annotation consistency, the selected ontology terms are stored in the database and associated to the user and the identifier used in the data file. Thus, when creating other sources using similar data and the same type identifier, the ontology terms will be preselected.

Mapping

The mapping of the data file fields into DAS concepts is one of the most important parts in the process of creating a source. The mapping interface, a table showing the data on the file plus a series of comboboxes, is flexible and allows the user to specify the relations as one-to-one, one-to-many or even many-to-many, where it makes sense.

For semantically specified file formats like GFF, a mapping taking that information into account is proposed. For other file formats, when data file fields have names -such as column names on tabular files-some simple pattern-matching heuristics are used to create a proposed mapping. Users can always change that proposal to adapt it to their specific needs.

Coordinate Systems

Another important concept in DAS is that of coordinate systems. They uniquely identify the sequences being annotated. In genomic DAS, for example, a coordinate system is defined by a species and assembly -i.e. we can specify that we are annotating the current genomic sequence for human by saying its the assembly “GRCh37” of “Homo sapiens”. While not required -there are cases when no suitable coordinate system is available, like when annotating a newly sequenced organism-, it is strongly recommended to specify the coordinate system for a new source. Most of the clients will refuse to integrate sources without a coordinate system, since it’s not possible to ensure that the annotations are referring to the same sequence. easyDAS provides a simple coordinate system selector with all the coordinates systems available on the DAS registry. It is possible to filter by any of the fields to help finding the right coordinate system for the source.

Source Creation

The source creation functionality is based on a wizard-like interface with a series of simple steps that guide the user through the process of defining the new source. At every step as much information as possible is extracted from either the data file or the context to reduce users’ work and decisions to the minimum.

The creation process starts with the file Upload form where a simple file selector field is available. While the user can specify the format of the uploaded file, some heuristics are in place to detect the format automatically. This will usually be the best option.

The second step in the source creation wizard shows a preview of how the file has been identified and parsed and some additional options to further refine the parsing (i.e. removing quotation, defining headers...). In case the automatic format identification mechanism fails, it is possible to amend the format selection in this second step (Figure 4).

The basic source metadata is gathered in the next step. The identifier of the source (which will be part of the source URL), its title, description and maintainer are specified here. The source coordinates system dialog is accessible from this step too.

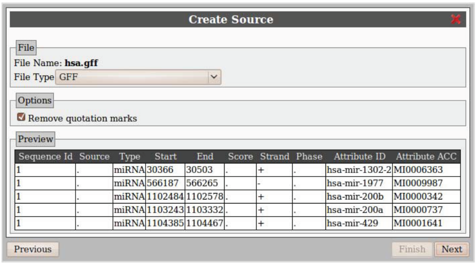
The fourth step is the mapping form. In this step, the user is required to link the data fields present in the uploaded data file to the standard DAS fields. This is the description of the data required to transform the data in the file into the DAS format. This is the last required step and it is possible to finish the wizard at any point from here.

To help in the simplification of the data files, it is not necessary to specify all the data fields for every feature in the file. The defaults form allows the user to define default values for absent fields or even for partially filled ones. It would be possible, for example, to specify the same method, type or score for all the features at once, maintaining those DAS concepts out of the primary data files.

The last two steps are used to specify the semantic annotations of the features. The ontology browser can be used to select the terms best describing the types and methods of the features in the file.

Data Storage and Access

The easyDAS interface is available at different address to the web interface and can be queried using standard



Create Source

File

File Name: **hsa.gff**

File Type: GFF

Options

☑ Remove quotation marks

Preview

Sequence Id	Source	Type	Start	End	Score	Strand	Phase	Attribute ID	Attribute ACC
1	.	miRNA	30366	30503	.	+	.	hsa-mir-1302-2	MI0006363
1	.	miRNA	566187	566265	.	-	.	hsa-mir-1977	MI0009987
1	.	miRNA	1102484	1102578	.	+	.	hsa-mir-200b	MI0000342
1	.	miRNA	1103243	1103332	.	+	.	hsa-mir-200a	MI0000737
1	.	miRNA	1104385	1104467	.	+	.	hsa-mir-429	MI0001641

Figure 4 File format description. After the data file has been successfully uploaded, this dialog allows to fine tune the file parser. At any moment, a preview of the parsed data is shown.

DAS requests. From a DAS point of view, each registered user has his own DAS server and his sources are created in that server. For example a user with a server name john would have a source called dataset1 available at http://www.ebi.ac.uk/das-srv/easydas/john/das/dataset1 and a list of all his sources at http://www.ebi.ac.uk/das-srv/easydas/john/das/sources. All the DAS requests are served by a slightly customized ProServer [14] also running on the EBI systems.

A MySQL relational database is used to store the sources data and can be accessed by both the web interface and the DAS server.

Custom easyDAS instances

easyDAS is free open source software. Its source code can be downloaded and modified freely in the terms of the GNU LGPL license. Although the reference instance is running at the EBI, it also means that it is possible to make custom installs of easyDAS independent of the EBI servers. It would be possible, for example, for an organization to provide a custom easyDAS setup in their own network so its groups can publish their data from the organization servers. It would even be possible to install easyDAS inside a private network and setup the included ProServer to reply only to requests coming from inside the network. That could be useful for organizations working with sensitive data but willing to share data between their own groups.

Conclusions

We have developed a system for the automatic creation of DAS sources. Users can upload data files with sequence annotations in different formats and define and create a new DAS source via a simple web-based wizard. Sources data will be stored on the EBI systems and freely available through a standard DAS interface. Data uploaded to easyDAS can then be easily integrated with other data on DAS using any of the available DAS clients such as Ensembl or Dasty2. easyDAS DAS sources are completely public and no access restrictions of any kind are applied.

As of today, using easyDAS is the easiest and fastest way of sharing a small or medium datasets over the DAS network. We think that this ease will encourage researchers with novel and unavailable datasets to publish and share them increasing the total amount of biological information available to the scientific community. Additionally, easyDAS will help those who need to share biological sequence annotations but can not run their own DAS server.

Implementation

easyDAS has two different entry points to the system: the web interface and the DAS server. The web interface is a client-server application, with the client being a web application written in Javascript and the server a set of cgi Perl scripts. The web interface is in charge of uploading and parsing the users datasets and offers the interface to define and manage the DAS sources. The other entry point, the DAS server, is a standalone ProServer instance with some minor modifications and is the one responding to the actual DAS commands to access to the data. A MySQL relational database accessible by both the web interface server side and ProServer stores the actual data (Figure 5).

Web interface

The web interface has two different parts, the client and the server. Communication between those parts is based on AJAX calls from the client to the server with JSON as the transport format.

Client

The web interface client part is a Rich Internet Application (RIA) managing the user interaction with the system. It has been written in Javascript and takes advantage of the object oriented nature of the language. The client uses the jQuery library to ease the DOM manipulation and to overcome the cross-browser

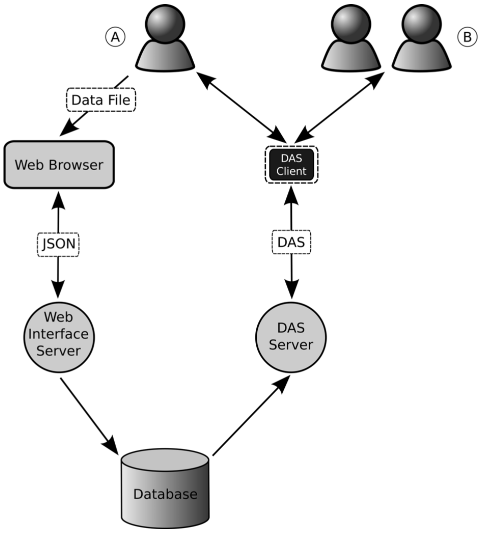


Figure 5 Implementation overview. A user (A) can access easyDAS using his web browser and upload a data file. The the easyDAS server, using the users description of the file, will extract the its data and insert it into the database. Anyone (B) can then access that information through the easyDAS DAS server using any of the available DAS clients, which are not part of the easyDAS system.

compatibility issues. While offering a rich interactive user interface, easyDAS' UI needs were simple and so the interface is custom built and only minimal parts of the jQueryUI framework are used (i.e. dialog dragging functionality).

Server

The server side of the web interface has been implemented as a set of Perl CGI scripts running behind an Apache server. A custom easyDAS module provides the actual functionality and defines a basic file parser class. The different parsers are specializations of that class and completely independent of other parsers, so adding a new file parser to the system is quite straightforward and does not require the modification of any other file on the system.

Database

The data back-end for easyDAS data is a MySQL relational database. The database is used to store both the information regarding the easyDAS system -such as which users are registered and what sources are available- and the actual biological data, the content of the DAS sources. The database schema was designed with the goal of isolating the sources as much as possible and so it has a set of six small tables for every source. This table setting maps very well to the hydra capability of the underlying DAS server and reduces the amount of code needed to implement the multiheaded DAS server. In addition, it simplyfies data insertion -no concurrent writes can happen on the same table- and deletion of sources -only whole table drops are required. Although the performance requirements of the database are low at the moment, it would be relatively easy to span the data over multiple database servers if it was needed.

This database is also accessible from the DAS server and so is the link between the two sides of the system.

DAS Server

The DAS server in easyDAS is a ProServer with a minor customization. ProServer, written in Perl and already extensively used in the EBI systems, offers both the power and efficiency required to potentially serve hundreds or thousands of DAS sources.

One feature that sets ProServer apart from other DAS servers is its capability to create multi-headed DAS sources, what it calls a hydra source. Hydra sources do not require specific per-source configuration as any other source type, but can be created on the fly by a Perl function -which in easyDAS is querying the database- based on a single basic configuration. Without hydras, either the server must be restarted on every source modification or many individual servers should have been created, requiring much more server power than the single ProServer instance in use. While it is true that other approaches would have been possible, none of them is so simple and yet powerful as using ProServer hydras.

Availability and requirements

- **Project name:** easyDAS
- **Project home page:** http://code.google.com/p/easydas/
- **Operating system(s):** Platform independent (web based)
- **Programming language:** Javascript, Perl
- **Other requirements:** none
- **License:** GNU LGPL
- **Any restrictions to use by non-academics:** none

Acknowledgements
BGM is supported by an FI predoctoral grant from Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya and Spanish MEC/MICINN under project LogicTools2 (TIN2007-68093-C02-01). XMP is supported by the project SweetLogics (TIN2010-21062-C02-01) funded by the Spanish Ministry of Science and Technology. RJ is supported by the European Commission FP6 "Enfin" grant, contract number LSHG-CT-2005-518254.

Author details
[1]Software Department, UPC-BarcelonaTech, Barcelona, Spain. [2]European Bioinformatics Institute, Hinxton, Cambridge, UK.

Authors' contributions
AJ, RJ and HH conceived the original idea of the system. BGM, AJ and RJ designed the system and BGM implemented it. AJ helped with the DAS server implementation. XMP and HH supervised the work. BGM drafted the manuscript. All authors read and approved the final manuscript.

Received: 18 May 2010 Accepted: 18 January 2011
Published: 18 January 2011

References

1. Parkinson H, Kapushesky M, Kolesnikov N, Rustici G, Shojatalab M, Abeygunawardena N, Berube H, Dylag M, Emam I, Farne A, Holloway E, Lukk M, Malone J, Mani R, Pilicheva E, Rayner TF, Rezwan F, Sharma A, Williams E, Bradley XZ, Adamusiak T, Brandizi M, Burdett T, Coulson R, Krestyaninova M, Kurnosov P, Maguire E, Neogi SG, Rocca-Serra P, Sansone S, Sklyar N, Zhao M, Sarkans U, Brazma A: **ArrayExpress update-from an archive of functional genomics experiments to the atlas of gene expression.** *Nucleic Acids Research* 2009, , **37 Database:** D868-D872.
2. Barrett T, Suzek TO, Troup DB, Wilhite SE, Ngau W, Ledoux P, Rudnev D, Lash AE, Fujibuchi W, Edgar R: **NCBI GEO: mining millions of expression profiles-database and tools.** *Nucleic Acids Research* 2005, , **33 Database:** D562-566.
3. Jenkinson AM, Albrecht M, Birney E, Blankenburg H, Down T, Finn RD, Hermjakob H, Hubbard TJP, Jimenez RC, Jones P, Kähäri A, Kulesha E, Macías JR, Reeves GA, Prlić A: **Integrating biological data-the Distributed Annotation System.** *BMC Bioinformatics* 2008, **9(Suppl 8)**:S3.
4. Dowell RD, Jokerst RM, Day A, Eddy SR, Stein L: **The distributed annotation system.** *BMC Bioinformatics* 2001, **2**:7.
5. **BioDAS.** [http://www.biodas.org/].
6. **PSICQUIC.** [http://code.google.com/p/psicquic/].
7. Hubbard TJP, Aken BL, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L, Coates G, Fairley S, Fitzgerald S, Fernandez-Banet J, Gordon L, Graf S, Haider S, Hammond M, Holland R, Howe K, Jenkinson A, Johnson N, Kahari A, Keefe D, Keenan S, Kinsella R, Kokocinski F, Kulesha E, Lawson D, Longden I, Megy K, Meidl P, Overduin B, Parker A, Pritchard B, Rios D, Schuster M, Slater G, Smedley D, Spooner W, Spudich G, Trevanion S, Vilella A, Vogel J, White S, Wilder S, Zadissa A,

Birney E, Cunningham F, Curwen V, Durbin R, Fernandez-Suarez XM, Herrero J, Kasprzyk A, Proctor G, Smith J, Searle S, Flicek P: **Ensembl 2009.** *Nucleic Acids Research* 2009, , **37 Database:** D690-D697.

8. Jimenez RC, Quinn AF, Garcia A, Labarga A, O'Neill K, Martinez F, Salazar GA, Hermjakob H: **Dasty2, an Ajax protein DAS client.** *Bioinformatics (Oxford, England)* 2008, **24(18)**:2119-2121.
9. Stein LD, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich JE, Harris TW, Arva A, Lewis S: **The generic genome browser: a building block for a model organism system database.** *Genome Research* 2002, **12(10)**:1599-1610.
10. Waterhouse AM, Procter JB, Martin DMA, Clamp M, Barton GJ: **Jalview Version 2-a multiple sequence alignment editor and analysis workbench.** *Bioinformatics (Oxford, England)* 2009, **25(9)**:1189-1191.
11. Prlić A, Down TA, Hubbard TJP: **Adding some SPICE to DAS.** *Bioinformatics (Oxford, England)* 2005, **21(Suppl 2)**:ii40-41.
12. **PeppeR - Graphical 3D-EM DAS Client.** [http://biocomp.cnb.uam.es/das/PeppeR/index.html].
13. Messina DN, Sonnhammer ELL: **DASher: a stand-alone protein sequence client for DAS, the Distributed Annotation System.** *Bioinformatics* 2009, **25(10)**:1333-1334.
14. Finn RD, Stalker JW, Jackson DK, Kulesha E, Clements J, Pettett R: **ProServer: a simple, extensible Perl DAS server.** *Bioinformatics (Oxford, England)* 2007, **23(12)**:1568-1570.
15. **Dazzle.** [http://www.derkholm.net/thomas/dazzle/].
16. **MyDas.** [http://code.google.com/p/mydas/].
17. Cote RG, Jones P, Martens L, Apweiler R, Hermjakob H: **The Ontology Lookup Service: more data and better tools for controlled vocabulary queries.** *Nucleic Acids Research* 2008, , **36 Web Server:** W372-W376.
18. Prlić A, Down TA, Kulesha E, Finn RD, Kähäri A, Hubbard TJP: **Integrating sequence and structural biology with DAS.** *BMC Bioinformatics* 2007, **8**:333.
19. Recordon D, Reed D: **OpenID 2.0.** *Proceedings of the second ACM workshop on Digital identity management - DIM '06* Alexandria, Virginia, USA; 2006, 11 [http://portal.acm.org/citation.cfm?doid=1179529.1179532].
20. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M: **The Sequence Ontology: a tool for the unification of genome annotations.** *Genome Biology* 2005, **6(5)**:R44.
21. Reeves GA, Eilbeck K, Magrane M, O'Donovan C, Montecchi-Palazzi L, Harris MA, Orchard S, Jimenez RC, Prlic A, Hubbard TJP, Hermjakob H, Thornton JM: **The Protein Feature Ontology: a tool for the unification of protein feature annotations.** *Bioinformatics (Oxford, England)* 2008, **24(23)**:2767-2772.
22. Montecchi-Palazzi L, Beavis R, Binz P, Chalkley RJ, Cottrell J, Creasy D, Shofstahl J, Seymour SL, Garavelli JS: **The PSI-MOD community standard for representation of protein modification data.** *Nature Biotechnology* 2008, **26(8)**:864-866.

doi:10.1186/1471-2105-12-23
Cite this article as: Gel Moreno *et al.*: easyDAS: Automatic creation of DAS servers. *BMC Bioinformatics* 2011 **12**:23.

Part II

Visualisation

Chapter 5

Genomic Data Visualisation

Data visualisation can be defined as the creation of graphical representations of data with the aim of better communicating it, making it understandable, explorable, or easing the process of knowledge discovery. The human mind is an excellent data analysis machine capable of efficiently absorbing visual information and spotting trends and relations[7,8] so creating visual representations of data helps tapping into that power when trying to analyze and understand vast amounts of data.

Although data has been represented in graphical forms for a long time and we are all used to plot and read charts in our everyday life, generalized use of computers have brought new means of collecting, managing and specially representing amazing amounts of data. In recent years, in part due to the appearance of new technologies making it easier to create new and different visual representations of data and in part thanks to the availability of new, interesting and sometimes huge datasets to work with, new visualisation techniques and trends have emerged. These new techniques and technologies have enabled the representation of previously unmanageable datasets and have brought us possibility of looking at data from a different point of view.

Biology, and specially genomics -and its many -omics siblings- have been one of the fields providing new data and challenges to the data visualisation experts. Visualisation of large biological datasets has been done using adapted and updated versions of classical techniques -e.g. volcano plots, heatmaps, dendograms... - or by creating new representation paradigms specifically tailored to the data being visualized. Two examples of these new paradigms are synteny maps, for the representation of the similarity between different genomes, and genome browsers, interactive representations of a genome with usually drawn along different types of genome anchored data.

Biological data representation is now a topic of great interest with even special

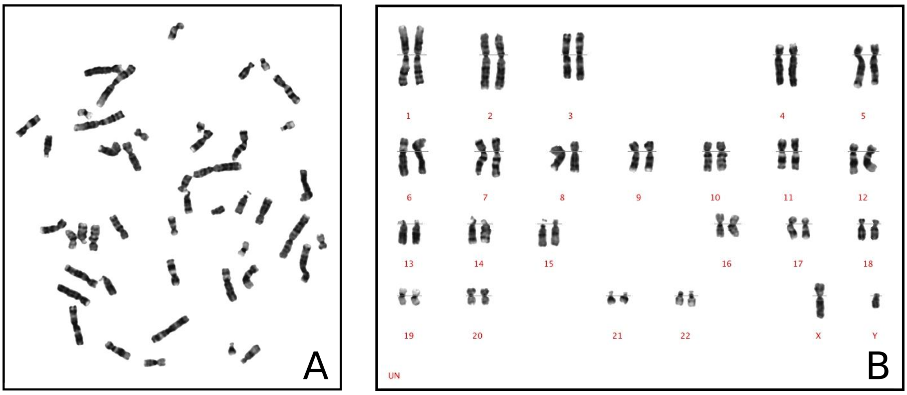


Figure 5.1 Example of a karyotype. This figure shows an example of what a genome looks like when seen through a microscope. A) The chromosomes of a single cell during the metaphase. B) The same chromosomes after being classified in the karyotype. We can see a normal male karyotype, with 44 autosomal chromosomes and 2 sex chromosomes, X and Y.

issues of important journals devoted to it[62] and international conferences such as VizBi[63] covering every aspect of it.

5.1 Genome Visualisation

The usual way to view a genome is in the form of a karyotype, with the DNA packed into chromosomes and those arranged in a predefined order (usually by size). It is relatively easy to take a picture of the stained chromosomes during the metaphase and arrange them to get an image of the genome (Figure 5.1). This image of the genome has been used to detect big structural changes such as trisomies, monosomies and other aneuploidies and big translocations (Figure 5.2). It is, however, a very coarse view of a genome and so it's being superseded in many applications by higher resolution techniques for *molecular karyotyping* such as aCGH and SNP array.

The karyotype images, also referred to as ideograms, have been frequently used as a model for genome representations, representing the different regions of the chromosomes, the bands, in different shades of gray. The NCBI MapView tool[64] or Ensembl's karyoview[65] are good examples of that. In addition, many tools are available to create karyotype representations that include our custom data[66 67 68 10], some of them taking that genomic representation to a next level with a circular representation of the genome and highly customizable data plotting options such as Circos[69] (Figure 5.3). All these tools and representations are mainly suitable for the representation of large features -translocations, copy num-

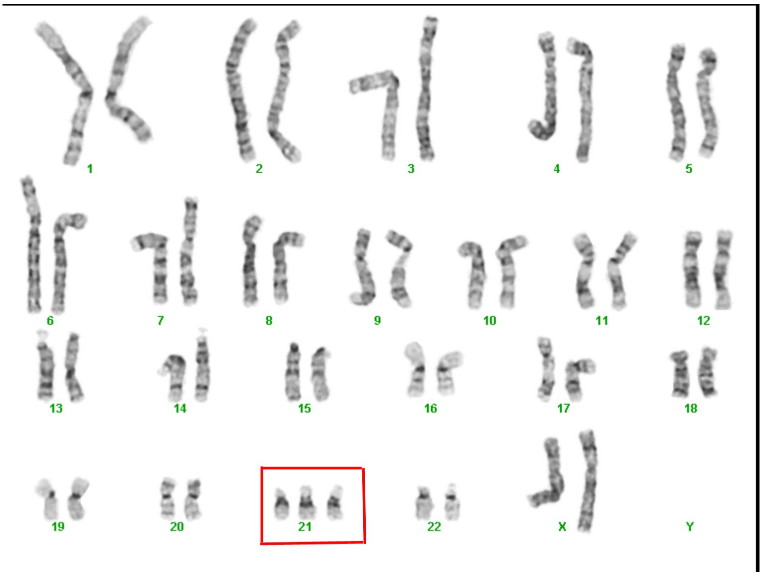


Figure 5.2 Example of altered karyotype In this example, we can see a clear trisomy of the chromosome 21.

ber gains or losses-, the approximate position of a small number of features -the position of 30 genes- or the aggregation of a higher number of features -gene density across the genome- or almost base level data -nucleosome occupancy across the genome-, and help in the identification of broad patterns. However, this kind of global views is not well suited for the study of smaller features.

In order to precisely represent much smaller features and their spatial relations -i.e. is this SNP located inside an exon? does the breakpoint of that translocation disrupt a gene?- other representations capable of representing a genome region at a base level are needed. And what about the relations between big and small features? and the exact boundaries of big regions? and being able to relate unique features and their aggregates? For this we need a tool able to show both a genome -or at least a chromosome- and go deep into it until base level resolution, from a karyotype up to a handful of nucleotides, and this is a big zoom range. If a the human chromosome 1, which comprises about 250 million bases is represented in a computer screen and in the same space we have to represent about 125 bases -a size good enough to comfortably explore data at a base level- we get a zoom range of about 20.000.000x. If we were representing the earth, that zoom range would bring us from the whole globe -with a diameter of 12600km- to a mere 6.4m taking the whole screen, which is about 60 times closer than the closest zoom we can get with Google Maps, and probably enough to read a license plate.

Genomic Browsers are able to represent genomes and their associated data in a meaningful, practical and efficient way.

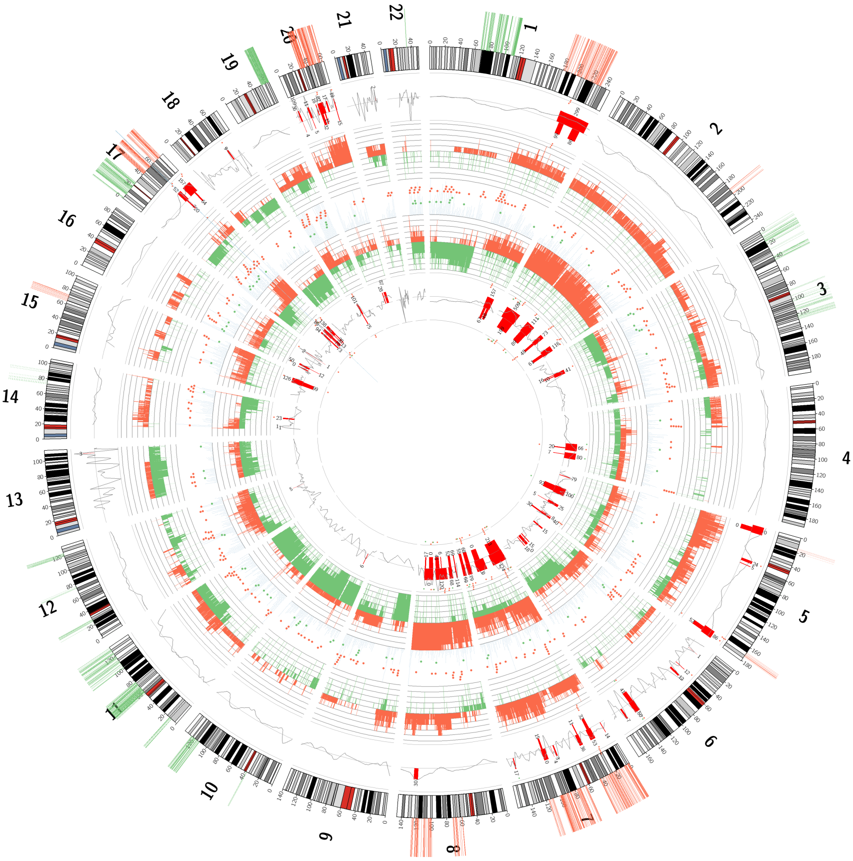


Figure 5.3 Circos plot. An example of the representation of the human genome and associated experimental data. Chromosome ideograms are arranged in a circle and data is plotted inside and outside of the that circle. Experimental data represents the gene expression patterns and copy number status of Malignant Peripheral Nerve Sheath Tumors (MPNSTs) from NF1 patients analyzed using different techniques.

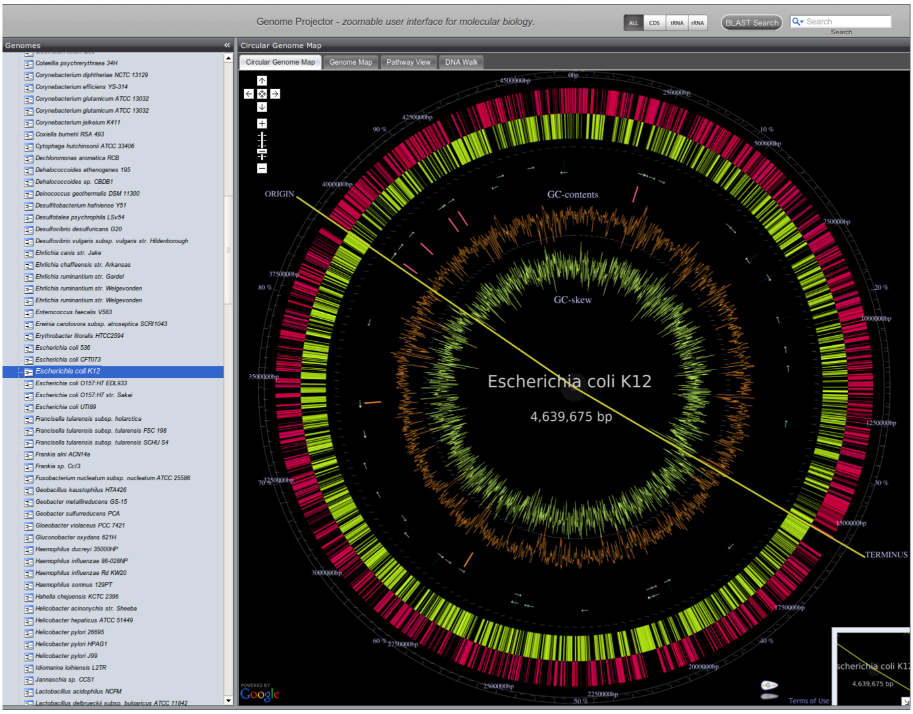

Genome Projector - zoomable user interface for molecular biology.
ALL
CDS
tRNA
rRNA
BLAST Search
Search
Genomes
Colwellia psychrerythraea 34H
Corynebacterium diphtheriae NCTC 13129
Corynebacterium efficiens YS-314
Corynebacterium glutamicum ATCC 13032
Corynebacterium glutamicum ATCC 13032
Corynebacterium jeikeium K411
Coxiella burnetii RSA 493
Cytophaga hutchinsonii ATCC 33406
Dechloromonas aromatica RCB
Dehalococcoides ethenogenes 195
Dehalococcoides sp. CBDB1
Deinococcus geothermalis DSM 11300
Desulfitobacterium hafniense Y51
Desulfotalea psychrophila LSv54
Desulfovibrio desulfuricans G20
Desulfovibrio vulgaris subsp. vulgaris str. Hildenborough
Ehrlichia canis str. Jake
Ehrlichia chaffeensis str. Arkansas
Ehrlichia ruminantium str. Gardel
Ehrlichia ruminantium str. Welgevonden
Ehrlichia ruminantium str. Welgevonden
Enterococcus faecalis V583
Erwinia carotovora subsp. atroseptica SCRI1043
Erythrobacter litoralis HTCC2594
Escherichia coli 536
Escherichia coli CFT073
Escherichia coli K12
Escherichia coli O157:H7 EDL933
Escherichia coli O157:H7 str. Sakai
Escherichia coli UTI89
Francisella tularensis subsp. holarctica
Francisella tularensis subsp. tularensis FSC 198
Francisella tularensis subsp. tularensis SCHU S4
Frankia alni ACN14a
Frankia sp. CcI3
Fusobacterium nucleatum subsp. nucleatum ATCC 25586
Geobacillus kaustophilus HTA426
Geobacter metallireducens GS-15
Geobacter sulfurreducens PCA
Gloeobacter violaceus PCC 7421
Gluconobacter oxydans 621H
Haemophilus ducreyi 35000HP
Haemophilus influenzae 86-028NP
Haemophilus influenzae Rd KW20
Haemophilus somnus 129PT
Hahella chejuensis KCTC 2396
Helicobacter acinonychis str. Sheeba
Helicobacter hepaticus ATCC 51449
Helicobacter pylori 26695
Helicobacter pylori HPAG1
Helicobacter pylori J99
Idiomarina loihiensis L2TR
Jannaschia sp. CCS1
Lactobacillus acidophilus NCFM
Lactobacillus delbrueckii subsp. bulgaricus ATCC 11842
Circular Genome Map
Circular Genome Map
Genome Map
Pathway View
DNA Walk
0bp
250000bp
500000bp
750000bp
1000000bp
1250000bp
1500000bp
1750000bp
2000000bp
2250000bp
2500000bp
2750000bp
3000000bp
3250000bp
3500000bp
3750000bp
4000000bp
4250000bp
4500000bp
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
ORIGIN
TERMINUS
GC-contents
GC-skew
Escherichia coli K12
4,639,675 bp
POWERED BY Google
Terms of Use

Figure 5.4 Screenshot of Genome Projector[72]. Genome Projector is a zoomable circular genome browser based on the G-Language[73] and the Google Maps technology, displaying the genome of *Escherichia coli*.

5.1.1 Genomic Browsers

There are two main classes of genome browsers, mimicking the two different types of genomes: circular and linear. There are a few circular genome browsers[70,71,72] and are optimized for the small circular genomes usually found in bacteria (Figure 5.4). It is not possible to use them to visualize large linear genomes and usually have a limited zoom range.

The other class of genome browsers expect a linear genome organized in one or more chromosomes. In linear genome browsers, the coordinate system is one-dimensional and corresponds to the base positions in the chromosome, and any additional data and genome annotations are displayed based on this coordinate system. In general, genome browsers organize genome annotations in tracks, with different data types drawn in it's specific space along the chromosomal coordinate marker. As opposed to most of the zoomable interfaces available, including mapping software and even circular genome browsers, linear genome browser implement one-dimensional zooming[74] and so when zooming only the chromosomal scale changes, not the dimension controlling the space available to each track. Most of them also incorporate some kind of semantic zooming[75], by which the level of detail of the data is automatically adapted to the zoom-level. For example, when visualizing SNPs, the user will be interested in the exact position of each SNP in very close zooms, but SNP density will be the representation of choice at broader zoom levels, since it wouldn't make any sense to represent the exact position of a million of SNPs in a thousand pixels representation of a big genome region.

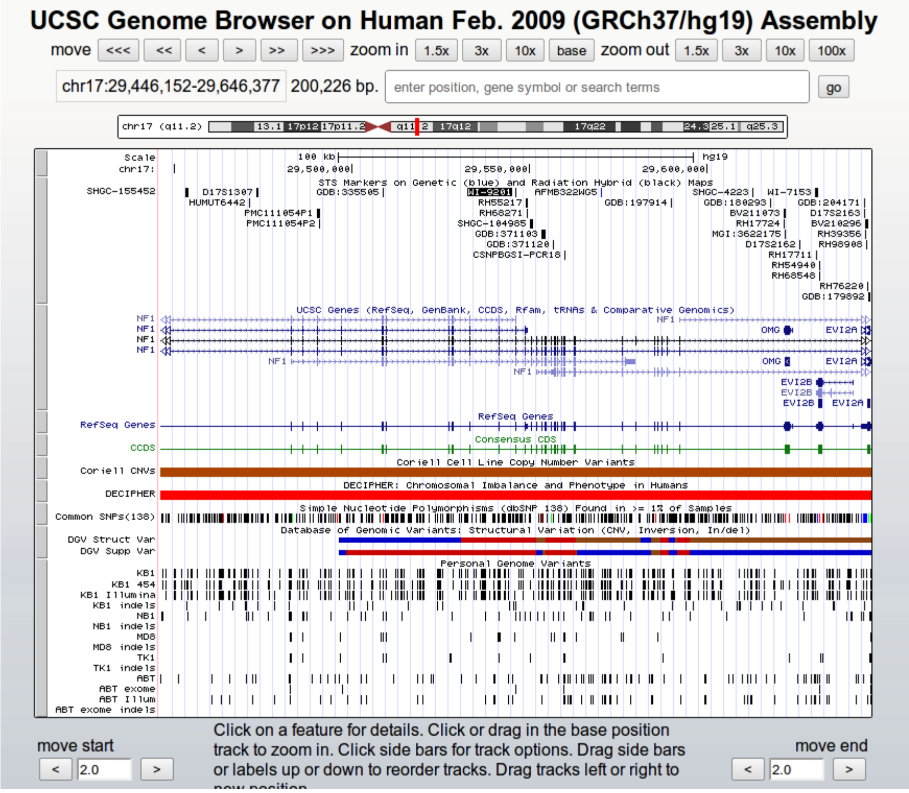
A genome browser needs to be able to access to huge databases of genomic data and create visual representations of that data. Due to the fast evolution of genomic databases and since images are much easier to transmit over the networks than the original datasets they represent, genome browsers were a good candidate to move to the web. Actually, in one of the first references to a genome browser, from 1996, Heumann *et al.*[76] present a web-based genome browser capable of displaying different types of annotations along the genome in a form with some of the characteristics of modern genome browsers. However, web-based genome browsers had an important problem: very limited interactivity. On the other hand, stand-alone genome browsers had a much better interactivity options, not being limited by HTML, but had to deal with data distribution somehow. Examples of stand alone genome browsers are IGB[77,78] -at one point one of the most versatile genome browsers-, IGV[79] -specialized in NGS data-, Apollo[80,81] -that includes data editing features in addition to visualisation-, Argo[82], and others[83].

Genome visualisation is still somehow an unsolved problem. There are many options available, very good browsers have been developed but new improvements are being regularly made. Lately, three reviews of the available genome visualisation tools have been published[84,85,25] and even a framework for the evaluation and comparison of genome browsers have been proposed[86].

5.1.2 Genomic Data visualisation on the web

Genomic data visualisation in the web has been limited by the possibilities of web technologies. HTML for representation, full HTTP requests for data transmission and web forms and simple javascript to provide basic interactivity was all that was available to web developers a few years ago. Any of these was conceived neither for the transmission nor the representation of the large amounts of data that genome browsers would require. The obvious choice was to exploit the server-side capacities to produce static images that were sent to the user to be drawn as part of a web page. This was the approach of Heumann's genome browser in 1996 and was the same used by the three main genome browsers: UCSC Genome Browser[87], Ensembl[13] and GBrowse[11].

For years, while the actual representation of the data improved, with cleaner and more diverse and specialized representations, and while data access evolved, adding the possibility of retrieving data from distributed sources via DAS or other means, the interactivity options had not changed much. As of 2007, the most widely used genome browsers, UCSC Genome Browser, Ensembl and those based


UCSC Genome Browser on Human Feb. 2009 (GRCh37/hg19) Assembly
move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x
chr17:29,446,152-29,646,377 200,226 bp.
enter position, gene symbol or search terms
go
chr17 (q11.2)
Scale
chr17:
100 kb
hg19
29,500,000
29,550,000
29,600,000
STS Markers on Genetic (blue) and Radiation Hybrid (black) Maps
UCSC Genes (RefSeq, GenBank, CCDS, Rfam, tRNAs & Comparative Genomics)
NF1
OMG
EVI2A
EVI2B
RefSeq Genes
CCDS
Consensus CDS
Coriell CNVs
Coriell Cell Line Copy Number Variants
DECIPHER
DECIPHER: Chromosomal Imbalance and Phenotype in Humans
Common SNPs(138)
Simple Nucleotide Polymorphisms (dbSNP 138) Found in >= 1% of Samples
DGV Struct Var
DGV Supp Var
Database of Genomic Variants: Structural Variation (CNV, Inversion, In/del)
Personal Genome Variants
KB1
KB1 454
KB1 Illumina
KB1 indels
NB1
NB1 indels
MD8
MD8 indels
TK1
TK1 indels
ABT
ABT exome
ABT Illum
ABT exome indels
move start
< 2.0 >
move end
< 2.0 >

Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position.

Figure 5.5 The UCSC Genome Browser. Even in 2014, the main method of interaction of the user with the UCSC Genome Browser is the button bar above the genome representation.

on GBrowse, had essentially the same navigation options that had been available for the last ten years: a button based interface to zoom in and out and navigate up a down of the genome. While combined with semantic search and the possibility of jumping anywhere of the genome by feature name or coordinate was enough to make genome browsers extremely useful, this approach did not encourage the exploration of the genome nor was the interactivity level users were growing used to when using others services (Figure 5.5).

There were attempts to break out of the rigidity imposed by HTML and offer greater interactivity, such as the one from Helt *et al.*[75] with BioViews, that presented a rich and interactive interface with on-demand data access based on Java and implemented as a Java applet. But at the end the Java applet technology suffered from its own problems and limitations and none of the major browsers adopted it.

When this project started there was a complete lack of highly interactive genome browsers, but the launch of Google Maps the year before showed that it was possible to create a web application capable of highly interactive direct manipulation of image sets. It was then when we started working on GenExp as a proof of concept to investigate the feasibility of a highly interactive web-based genome browser.

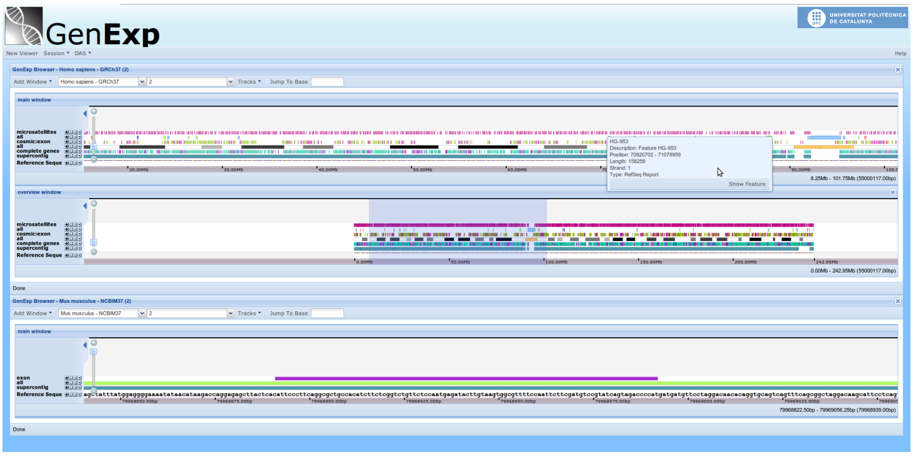
In recent years, internet browser technology has improved dramatically both in term of speed and capabilities and this have brought a host of new web-based libraries and tools. The improvement of web technologies has had an impact in the field of genome browsers too, and new AJAX and HTML5 based genome browsers have been presented. In addition to GenExp, our technology prototype genome browser[29,28], other fully web based genome browsers have appeared such as JBrowse[88] -an interactive genome browser with client side data rendering-, myKaryoView[89] -a genome browser and DAS client with client side data rendering based on HTML elements- and Dalliance[90] -an interactive genome browser and DAS client with client side data rendering based in SVG but limited to displaying region of at most 500kb-. In addition, genome browser toolkits, libraries designed to help in the implementation of domain or organisms specific genome browsers have been developed, such as: the UTGB Toolkit[91], capable of acting as a DAS client but offering no direct interaction with the genome and maintaining the traditional button based navigation and Scribl[92], a web-based genome browser toolkit exploiting the same technological approach from GenExp, an HTML5's canvas-based client side rendering of genomic data.

Chapter 6

GenExp: an interactive genomic DAS client with client-side data rendering

Genexp is a web-based highly interactive genome browser designed to study and test the feasibility of new technological approaches to genomic data visualization in the web as well as different means of interaction with the genome representation (Figure 6.1). It was one of the first genome browser to take advantage of the new graphical capabilities of modern web browsers implementing client-side data rendering. With this approach, the client does not need to contact with the server for every redraw event and so there is no lag when panning and or performing small changes in the zoom level. Thanks to this, it was possible to implement a direct interaction approach to the genome manipulation and move from the standard button-based interactivity to a more natural mouse-based interaction. GenExp is a DAS client, and all its data is retrieved from distributed DAS servers, with no local database whatsoever.

The GenExp project started in the fall of 2006, a year after Google Maps was presented and when, to our knowledge, no other highly interactive genome browser was available. With the new graphical capabilities of web browsers implemented just a few months before that and with the promised speed improvements in client-side technologies, it seemed that it was the right time to investigate the possibilities of these new technologies and its applicability in the visualization of genomes. Nowadays, new genome browsers have been created, and some of them[93][94] have implemented some of the ideas we proposed with GenExp.


GenExp
UNIVERSITAT POLITÈCNICA DE CATALUNYA
New Viewer
Session
DAS
Help
GenExp Browser - Homo sapiens - GRCh37 (2)
Add Window
Homo sapiens - GRCh37
2
Tracks
Jump To Base
main window
microsatellites
all
cosmic:exon
all
complete genes
supercontig
Reference Seque
20.00Mb
30.00Mb
40.00Mb
50.00Mb
60.00Mb
90.00Mb
8.25Mb - 101.75Mb (55000117.00bp)
overview window
0.00Mb
50.00Mb
100.00Mb
150.00Mb
200.00Mb
242.95Mb
0.00Mb - 242.95Mb (55000117.00bp)
Done
GenExp Browser - Mus musculus - NCBIM37 (2)
Mus musculus - NCBIM37
exon
supercontig
agccatttatggaggggaaaatataacataagaccaggagagcttactcacattcccttcaggcgctgccacatcttctcggtctgttctccaatgagatacttgtaagtggcgttttccaattcttcgatgtccgtatcagtagaccccatgatgatgttcctaggacaacacaggtgcagtcagtttcagcggctaggacaagcattcctcag
79968850.00bp
79968875.00bp
79968900.00bp
79968925.00bp
79968950.00bp
79968975.00bp
79969000.00bp
79969025.00bp
79968822.50bp - 79969056.25bp (79968939.00bp)
Show Feature

HG-953
Description: Feature HG-953
Position: 70920702 - 71078959
Length: 158258
Strand: 1
Type: RefSeq Report

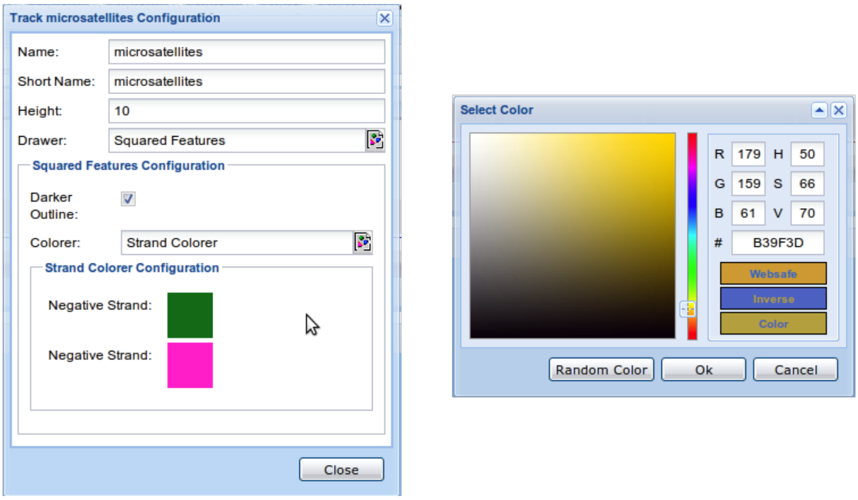
Figure 6.1 GenExp screenshot. In this image we can see a screenshot of the genome browser GenExp. In this session the user has three genomic views: two linked ones representing different zoom levels of the same region of the human genome and an independent one respresenting a region in the mouse genome.

6.1 GenExp Overview

GenExp is a web-based genome browser. It retrieves genomic data and annotations from DAS servers and presents them as tracks annotating the genome. It offers a highly interactive interface with mouse driven direct interaction with the genome representation and fast panning and zooming. A general view of GenExp in action can be seen in figure 6.1.

The main feature of GenExp is its high interactivity. It is possible to simply drag the genome to move its representation window and to use the mouse wheel to zoom in and out. The response to these interactions events is usually fast and exploration is encouraged. GenExp also implements the expected "jump to a certain position" feature for fast and directed positioning. As a DAS client, all data is retrieved from DAS servers and it is possible to add any valid DAS source to it, meaning that merging custom data with other public DAS sources is easy. GenExp has a mechanism to change the drawing configuration in a per track basis. Using the track configuration dialogs it is easy to change how the tracks are drawn, including shapes, color schemes and colors (Figure 6.2).

GenExp also implements sessions. Instead of being tied to a particular user, in GenExp sessions are small JSON files with the exact definition of the state of the application. It is possible to store these files for future use, but it is also possible to share them with other users effectively sharing an the exact view of the genome and its annotations.



Track microsatellites Configuration

Name: microsatellites

Short Name: microsatellites

Height: 10

Drawer: Squared Features

Squared Features Configuration

Darker Outline: ☑

Colorer: Strand Colorer

Strand Colorer Configuration

Negative Strand: 🟩

Negative Strand: 🟪

Select Color

R	179	H	50
G	159	S	66
B	61	V	70

B39F3D

Figure 6.2 Track configuration. In GenExp it is possible to configure how each track will be drawn. In the configuration dialog (left) the user can select the drawer, the colorer and the different parameters. It is even possible to select a specific color with the color selector dialog (right).

Contrary to most available genome browsers, in GenExp is possible to have more than on genome representation at the same time. An arbitrary number of views can be active at any given time showing data from different genomes and displaying a different set of data tracks. A special kind of views are synchronized views. Synchronized views can be used to either extend the visible range or to have a different zoom level of the same region. It is possible, for example, to have three views showing a large region at the base level, while having a zoom window showing the same region at the gene level -few megabases- and a complete chromosome view to get the broad picture (Figure 6.1).

The main technological novelty of GenExp was the use of a client-side data rendering approach based on the new HTML5's canvas element. Instead of sending a request to the server with the necessary parameters -chromosome, start, end, data tracks, ...- and receiving an image in return, the GenExp client code asks for data, caches it, and creates the representation in the client machine. The next time a small change in the view is requested -e.g. move 200bp to the left-, the data is already in the client cache and no interaction with the server is needed at all.

GenExp has been implemented as a client-server application, with a powerful client written in javascript and a small server written in perl. The server side is mainly a thin layer with four missions: caching, translating, binning data and overcoming the same origin policy (SOP). The SOP prevents websites from connecting to servers others than the one they originate from and is enforced for security reasons. While in recent years it is possible to override this policy using the

Cross Origin Resource Sharing (CORS) options, when the GenExp project started the best way to connect to arbitrary servers was to use your own origin server as a proxy. It also translates the DAS responses from the original XML of DAS to a more compact, less verbose and javascript-friendly JSON structures. In addition to further improve performance both in network usage, client memory usage and drawing time, the server produces summarized density data. This data will be used to produce the plots corresponding to the broader zoom levels, where plotting individual features wouldn't be useful and is an implementation of semantic zooming. Finally, all data produced is cached in the server to reduce the queries to the original DAS servers.

The client part of GenExp is implemented as a modular event-based application and takes care of the interaction with the user, data management and rendering. The communication between the different elements of the user interface, the data managing code and the controller code is based in Prototype's implementation of custom events, since custom events were not natively supported at the time of writing GenExp. The user interface has been implemented with the ExtJS framework[31], a javascript library to create desktop-like responsive interfaces with the standard interaction elements such as menus, toolbars, sliders and dialogs.

The rendering is done using the new canvas element, first introduced by Apple in WebKit and later approved as part of the W3C HTML5 standard. Canvas provides a procedural drawing API to programmatically create bitmaps. As opposed to SVG and other vector based technologies, the resulting element not a collection of complex objects to be managed and redrawn but a single bitmap object completely indistinguishable from an image downloaded from the server. This approach allows for fast manipulation of the genome representation and together with some implementation tricks -the use of an `iframe` to sandbox the genome representation and not trigger DOM reflow events when panning and zooming-allowed us to implement fast and seamless panning even with the much slower javascript engines available in 2007.

To get the genomic information, GenExp uses the server proxy. After receiving new genomic data, the client creates the image using calls to drawing subsystem that in turn will call the canvas API and inserts it as part of a track into the genome viewer. Both the original data received and the image are cached, providing, together with the caching happening in the server, a three level cache system to GenExp. The cached raw data will be used in case a zoom event occurs and the zoom change is small enough as not to affect the semantic zooming level of detail. In addition, for custom DAS servers added by the user, GenExp uses jsDAS[36] to almost directly connect to the DAS servers using the server only as a proxy but skipping data transformation, translation and caching. It is useful specially for small DAS servers with user specific data to be shown together with genomic annotations, for example custom DAS servers created with easyDAS (see chapter 4).

6.2 Example Usage

We will illustrate a possible usage of GenExp building upon the easyDAS example developed in Chapter 4. In this part, we will see how GenExp can be used to explore the content of a DAS source and to create potential biological hipotesis.

6.2.1 Data Files

An additional data file with data retrieved from a database will be used in this example.

all.3k.promoters.csv This file contains data downloaded from the UCSC genome browser. It lists the 3kb regions before the beginning of the genes representing their promoter regions. The file is a tab separated file with six columns (`chromosome`, `start`, `end`, `gene symbol`, `type` and `identifier`) and contains 25458 records. This is an example of the first few lines of the file.

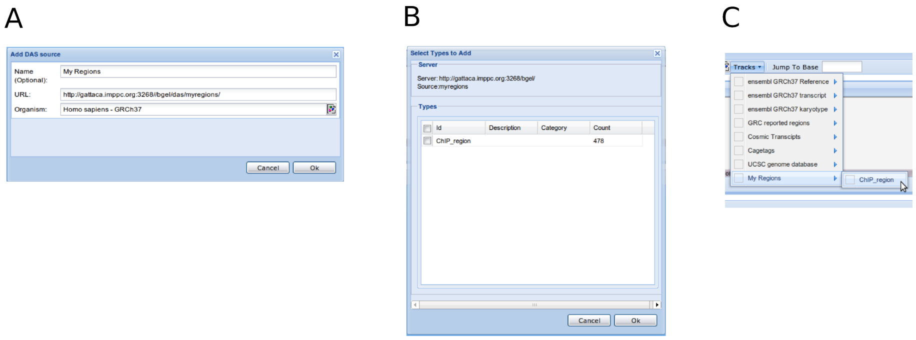
```
chr  start    end     name         type         id
1    8873     11873   DDX11L9      Promoter_3k  promoter_1
1    66090    69090   OR4F5        Promoter_3k  promoter_2
1    318083   321083  DQ597235     Promoter_3k  promoter_3
1    318145   321145  DQ599768     Promoter_3k  promoter_4
1    319036   322036  LOC100133331 Promoter_3k  promoter_5
1    324545   327545  LOC388312    Promoter_3k  promoter_6
```

6.2.2 Background

This example represents the continuation of the worflow started in the example in Chapter 6 about the study of the genomic regions obtained from a ChIP-seq experiment. After analyzing the data from the experiment to generate the genomic regions, our researcher created a DAS data source using easyDAS. He is now ready to explore the data with GenExp.

6.2.3 Example

To explore the data from a DAS source, connect to GenExp at http://gralggen.lsi.upc.edu/recerca/genexp/. Once there, select the organism, in our case ”*Homo sapiens - GRCh37*“, and a chromosome. Then, to add a new DAS source, open the *DAS* menu and select *Add DAS Source*. In the dialog (Figure 6.3 **A**), enter the URL of the DAS source created before, http://gattaca.imppc.org:3268//bgel/das/myregions/, give it a name and select an organism. After clicking *Ok*, a dialog will appear informing about the content of the DAS source, in our case 478 regions of the type *ChIP_region* (Figure 6.3 **B**), and asking what types to add. Select the



Name (Optional): My Regions

URL: http://gattaca.imppc.org:3268//bgel/das/myregions/

Organism: Homo sapiens - GRCh37

Server: http://gattaca.imppc.org:3268/bgel/
Source:myregions

□	Id	Description	Category	Count
□	ChIP_region			478

Figure 6.3 Screenshots of GenExp. **A**: Add source dialog. **B**: Type selector. **C**: The newly added DAS source is available at tracks menu.

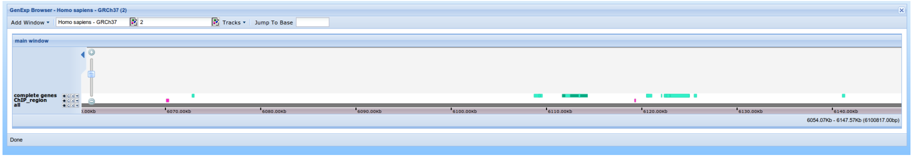


Figure 6.4 Screenshots of GenExp. The ChIP-seq regions are near the ends of genes.

ChIP_region type and click *Ok* again to add the DAS source to the list of available tracks.

Add the newly created track to the window by selecting it from the tracks menu (Figure 6.3 **C**). In order to have a clear reference of the chromosome add the ideogram (*Tracks - ensembl GRCh37 karyotype - all*) and add also the genes to study their positional relation with our ChIP-seq regions (*Tracks - ensembl GRCh37 transcript - complete genes*). You'll see nothing, since at the current zoom level there are no features in our window. You can use the mouse to navigate arround the chromosome: use the mouse wheel to zoom in and out and click and drag to move the chromosome. You can use the track control buttons (left of the window) to change the order and the appearance of the tracks. Exploring the data you will probably see that our ChIP-seq regions tend to be next to genes (Figure 6.4). To determine if they fall in promoter regions, we can created a new DAS source with easyDAS using the data downloaded from UCSC and add it to GenExp. After adding the the promoters track we can see that indeed, our ChIP-seq regions overlap the gene promoter regions (Figure 6.5) and so we could hipothetize that the original protein may have some function related to gene expression regulation.

In addition, we can study the distribution of our genomes on the whole genome. To do that, we can zoom out until we see the complete chromosome. With this whole chromosome view we can see that our regions are not randomly distributed

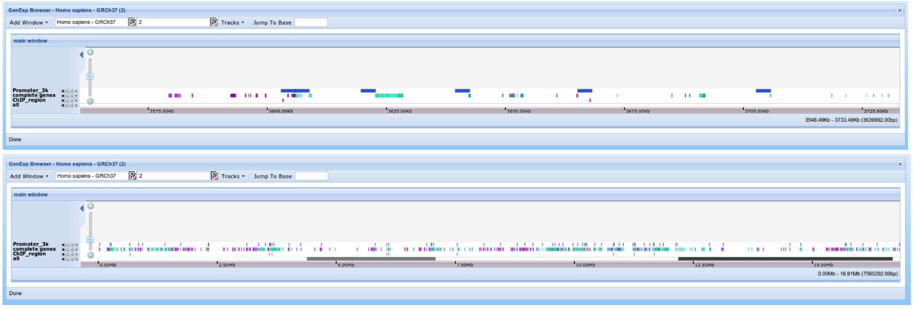


Figure 6.5 Screenshots of GenExp. Two views at different zoom levels of how ChIP-seq regions overlap with gene promoter regions.

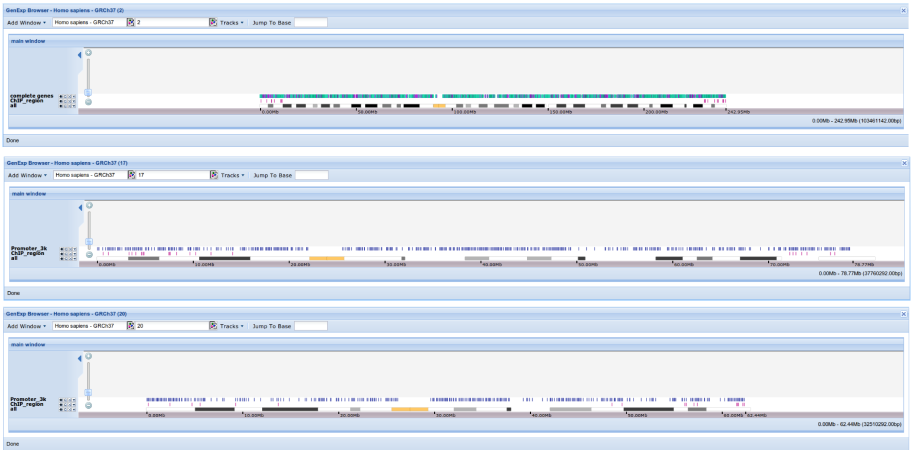


Figure 6.6 Screenshots of GenExp. Our ChIP-seq regions cluster near the telomeric regions. This can be seen in different chromosomes, in this case 2, 17 and 20.

on the chromosomes, but they cluster near the telomeric regions (Figure 6.6). We can change chromosomes and check that this clustering can be observed in all the chromosomes. To check wether these associations are statistically significant we should use other programs, such as the R package `regioneR`.

6.3 Key Contributions of GenExp

This is a summary of the key contributions of GenExp:

Interactive: GenExp is highly interactive. To our knowledge, at the time the project started no interactive web-based genome browser was available. GenExp proved that it was possible to create a highly interactive genome browser with fast response times with the web technologies and javascript engines available.

No arbitrary limits: As a DAS client, GenExp imposes no arbitrary limits on the accessible sources. It is possible to access any valid DAS source and data will be retrieved and displayed. In addition, in contrast with other interactive web-based genome browsers, no arbitrary limit is imposed in the length of the region to be shown, being possible to show from the base level to whole chromosomes.

Multiple Views: In GenExp it is possible to have any number of genome viewers at the same time. These viewers can be completely independent and represent different genomes or different parts of the same genome.

Linked Views: In addition to the independent genome views it is possible to have an arbitrary number of linked views either extending the represented region or centered at the same base but at a different zoom level.

Session management: GenExp implements session management. It is possible to save the application state in a session file either to save it for later use or to share the genome view with others.

Configurable track drawing: A simple track configuration option has been implemented and it is possible to change the appearance of a track including the glyphs and colors.

Client-side rendering using canvas: It uses the procedural API of the canvas element to render the genomic data without incurring in the object management overhead of SVG, VML and other vector based graphics. Client-side data rendering reduces the number of requests made to the server and allows for small changes in the representation parameters with no connection at all. Thanks to that, panning and small zooming are near instant.

6.4 GenExp publication

The paper presenting GenExp was published on PLOS One on June 2011. A preliminary version of the work titled *Implementing an Interactive Web-Based DAS Client*[29] was published on the Proceedings of 2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008) on 2009 published by Springer on their series Advances in Intelligent and Soft Computing.

This work has also been presented in various congresses and workshops either as a talk (*Bioinformatics Open Source Conference (BOSC 2009) - 17th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB*

2009) (Stockholm, Sweden, 2009), *DAS Workshop 2009* (Hinxton, United Kingdom, 2009) and *International Workshops on Practical Applications of Computational Biology and Bioinformatics 2008* (Salamanca, 2008)), a demonstration (*VIII Jornadas the Bioinformática* (Valencia, 2007)) or as a poster (*17th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB 2009)* (Stockholm, Sweden, 2009) and *X Jornadas the Bioinformática (Málaga, 2010)*)

Author contributions: XM and me conceived the project. With helpful suggestions and guidance from XM, I designed the system, implemented and tested it. The manuscript was written by me, and revised and approved by XM prior to publication.

The actual paper follows.

OPEN ACCESS Freely available online PLoS one

GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering

Bernat Gel Moreno*, Xavier Messeguer Peypoch

Software Department, UPC-BarcelonaTech, Barcelona, Spain

Abstract

Background: The Distributed Annotation System (DAS) offers a standard protocol for sharing and integrating annotations on biological sequences. There are more than 1000 DAS sources available and the number is steadily increasing. Clients are an essential part of the DAS system and integrate data from several independent sources in order to create a useful representation to the user. While web-based DAS clients exist, most of them do not have direct interaction capabilities such as dragging and zooming with the mouse.

Results: Here we present GenExp, a web based and fully interactive visual DAS client. GenExp is a genome oriented DAS client capable of creating informative representations of genomic data zooming out from base level to complete chromosomes. It proposes a novel approach to genomic data rendering and uses the latest HTML5 web technologies to create the data representation inside the client browser. Thanks to client-side rendering most position changes do not need a network request to the server and so responses to zooming and panning are almost immediate. In GenExp it is possible to explore the genome intuitively moving it with the mouse just like geographical map applications. Additionally, in GenExp it is possible to have more than one data viewer at the same time and to save the current state of the application to revisit it later on.

Conclusions: GenExp is a new interactive web-based client for DAS and addresses some of the short-comings of the existing clients. It uses client-side data rendering techniques resulting in easier genome browsing and exploration. GenExp is open source under the GPL license and it is freely available at http://gralggen.lsi.upc.edu/recerca/genexp.

Citation: Gel Moreno B, Messeguer Peypoch X (2011) GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering. PLoS ONE 6(7): e21270. doi:10.1371/journal.pone.0021270

Editor: Nicholas James Provart, University of Toronto, Canada

Received August 10, 2010; **Accepted** May 27, 2011; **Published** July 5, 2011

Funding: BGM was supported by an FI predoctoral grant from Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya (http://www.gencat.cat/diue/). BGM and XMP are also supported by the projects SweetLogics (TIN2010-21062-C02-01) and LogicTools-2 (TIN 2007-68093-C02-01) funded by the Spanish Ministry of Science and Technology. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: bgel@lsi.upc.edu

Introduction

In recent years the volume of genomic data on genome sequences and their annotations have been growing at a rapid pace. New organisms are sequenced every year, annotations on those sequences are constantly produced and refined and new data associated with these annotations is created. Most of this data is stored in public databases which are freely accessible and usually offer various options to browse and download the data, most of them via web interfaces. However, not all the databases offer programmatic access to their data nor a common format for its downloadable files. The DAS system provides a standard programmatic interface that is relatively easy to implement. This is specially important for small databases that lack the resources to develop and maintain such a system.

The Distributed Annotation System

In an attempt to address some of these issues the Distributed Annotation System (DAS) was proposed [1,2] (http://www.biodas.org). DAS is a client-server protocol designed to share and integrate annotations on biological sequences. It is based on standard web technologies, HTTP and XML, and offers a REST-like interface. DAS is currently widely used, with more than 1000 sources from more than 50 organizations, and some of the biggest public databases of biological data offer access to its contents via DAS [3]. The DAS Registry [4] offers a source discovery service with listings including most of the available sources. The DAS architecture was designed around the idea of having a small number of complex clients integrating data coming from many different simple sources.

DAS clients are responsible for multi-source data management, integration and representation. This means that DAS clients are usually complex applications. There are currently a number of stand-alone or web-based DAS clients available. Clients can be loosely classified into two different categories: protein-oriented clients are specialized in showing deep annotation of a relatively short sequence while genomic-oriented clients are capable of managing many annotations over a long sequence.

Protein-oriented clients usually offer a wide range of representation possibilities (3D structure, interactions graph, alignments) and most of them are stand-alone applications [5–9]. A notable exception is Dasty [10], a web-based protein-oriented client that creates an interactive and visually attractive representation of protein data in the browser without relying on the newest web technologies. It uses series of specially styled and positioned HTML div elements to create the graphical representation of the features and a Java applet to show the 3D structure of the protein. This approach, however, does not scale well to the data density needed in genomic-oriented clients.

Genomic data visualization

Due to its multi-scale nature, genomic data visualization is challenging. Genomic browsers have to deal with very long sequences (in the range of hundreds of millions of bases for some chromosomes) and manage their annotations in an efficient way. Since the sizes of features vary from one base to several megabases a very wide range of zoom levels is necessary: the whole chromosome representation needs to be a million times more dense than the base-pair view while remaining informative.

Stand alone genomic browsers [11,12] have full access to the underlying OS functionality and can take advantage of disc-based caching mechanisms and advanced drawing capabilities. On the other hand, web-based genome browsers are restricted to the web browser environment, and so, access to the underlying hardware is severely limited. No disc access is possible, memory management functions are not available and drawing is limited to HTML related technologies.

To overcome the limitations imposed by the web browser environment, most of the web-based genomic browser offload the data representation responsibilities to the server side, where static images are created to be later shown by the client running on the web browser. Ensembl [3], UCSC Genome Browser [13] and GBrowse [14] are examples of this approach. Creating the image on the server, however, usually implies a trade-off on interactivity since every change on drawing parameters (position, zoom level, active data set) results in a server request. Some attempts have been made to use tiling images, like Google Maps, but this approach has severe scalability problems given the extreme zoom range needed in genome visualization.

Client-side data rendering allows for greater interactivity, since minor changes on drawing parameters would not trigger a server request. This approach, however, has important constraints on available memory and rendering capabilities. KaryoDAS is a web-based client with client-side rendering that uses standard HTML entities to create its renderings. This imposes limitations on the number of features that can be drawn at the same time and its appearance and thus, KaryoDAS can only use DAS sources offering pre-filtering capabilities. Using newer web technologies such as HTML5 canvas it would be possible to build an interactive web-based genomic DAS client able to use any of the available DAS sources.

Goal

The goal of this work is to take advantage of the newest web technologies (specially the canvas element on HTML5) to create a new DAS genome browser with a fully interactive user interface with real-time zooming and panning. The new browser is based on web standards and does not need any browser plugin such as Flash or Java. It is open source and freely available.

Results and Discussion

Our genome browser GenExp -available at http://gralggen.lsi.upc.edu/genexp- offers a web-based ready-to-use interactive genome browsing experience and leverages the richness of the genomic DAS data sources with the ease of web applications.

Overview

As a genome oriented DAS client, GenExp is able to display data coming from the available genomic DAS sources in an integrated way, creating a single representation with data from different sources represented side-by-side and precisely positioned over its common reference sequence.

Data is drawn in tracks, each one representing a certain kind of data coming from a DAS source. Groups of tracks with data from the same region viewed at the same zoom level are drawn together in a track container. Dependent and synchronized track containers can be created to show an extended region, or to view the same region at different zoom levels. It is possible, for example, to study a region at a detailed base level and at the same time keep a view of its gene and chromosome contexts.

GenExp is fully interactive and its interaction scheme has been designed to encourage the exploration of the represented data. A view can be moved by dragging it with the mouse, centered by double-clicking on it and zoomed with the mouse scroll-wheel in the same way as it is done on map-viewing applications. Detailed information on a given feature can also be shown.

While GenExp is capable of rendering whole chromosomes at once, its interactive interface excels when not making drastic changes in zoom levels, since once the data is in memory, moving around a region has no network overhead. Since drawing times are in the order of milliseconds, any delay will be due to network latency. This means that relatively small movements and zoom changes will be performed almost immediately, incurring in no noticeable delay.

Since GenExp is a web-based application, no installation is needed: it is always ready to use via the web browser. Despite its web nature, it offers a desktop-like user experience by relying on common user interaction elements and widgets, such as menus, dialogs and toolbars.

Data Sources

A small list of sources is pre-configured in our public instance of GenExp and it is possible to add any genomic DAS source to GenExp simply entering its URL on the provided dialog. Using this feature it is possible to include even custom data by creating a new DAS source, for example using easyDAS [15] (http://www.ebi.ac.uk/panda-srv/easydas/), the automatic DAS server creation tool, and displaying it along the preexisting sources.

Multiple Views

GenExp can manage an arbitrary number of data containers at the same time. These multiple viewers can be dependent or independent.

Dependent viewers might be either different zoom levels of the same data, in order to get a broader context view when examining a close-up view of a region for example, or "next" and "previous" regions in order to extend the region displayed at the same zoom level.

Independent viewers can show data from any genomic region, even from other organisms, so it is possible to compare related regions of different genomes side-by-side (Figure 1).

Sessions

Session management capabilities are included in GenExp. It is possible to save the state of the application at any given moment to later rebuild it. Session info is stored in flat text files so it is possible to store and share them.

Customization

In addition to data customization, appearance tweaking is also in place for both the server administrator and final user. End users

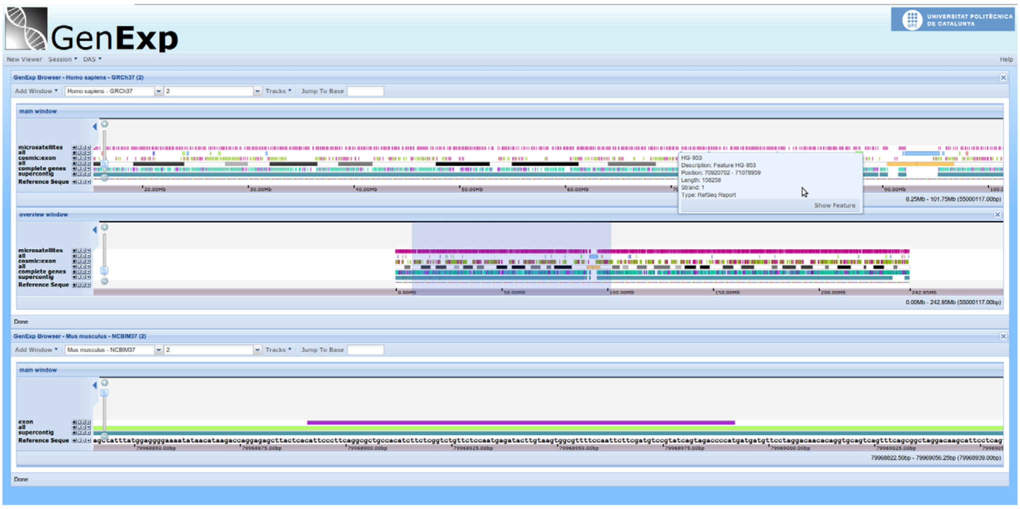


Figure 1. Main view of GenExp. Different regions from two organisms are represented at the same time. For human assembly GRCh37 there is a view showing more than 80 Mb and an additional linked Overview window. The other window is showing a region containing an exon at a base level view for the *Mus musculus* assembly NCBIM37.
doi:10.1371/journal.pone.0021270.g001

can customize the rendering process, selecting any of the available drawing and colouring tools or changing the individual colors used. These changes will take place instantly since no communication with the server is required. Administrators can define per-track defaults, available options, or even create new drawing routines with a few lines of code.

Limitations

GenExp has some limitations that should be noted. First of all, due to server and network capacity, requesting large regions of very dense tracks can result in timeouts or failed requests. There are some mechanisms in DAS to prevent this but unfortunately not all servers implement them. Additionally, some incompatibilities exist with some web browsers and so GenExp has been only proven to work completely on Firefox version 1.5 or newer. The functionality available on other browsers may vary.

Methods

Design and Implementation

While GenExp as a whole is a DAS client, the system itself has a client-server architecture with a complex web-based Javascript client talking to a simple Perl server (Figure 2). While the client is responsible for user interaction, data management and representation, the server is mainly a proxy with some basic caching capabilities.

The client. The client is a rich Internet application implemented in Javascript and runs inside the web browser. It is completely modular and easily modifiable and extensible.

Since GenExp creates the data representation on the client side, there is an important part of the client devoted to data gathering and management. Data is requested from the server only when needed and based on zoom level and viewing position, minimizing server load and network traffic. All received data is cached in memory so fewer requests are needed when moving and zooming.

The drawing strategy used by other genome browsers with client-side rendering is to use standard div elements to represent the genomic features. This approach has some advantages but has limitations of scalability specially of creativity, since customization possibilities are limited. GenExp, on the other hand, takes advantage of the latest HTML5 capabilities added to web browsers to create the representation of genomic features. The new canvas element exposes a low-level procedural drawing API so it is possible to create actual images on-the-fly and move them with no extra overhead. This means that when creating the representation, instead of creating a complete HTML element for every feature GenExp only draws a glyph on an image element. This has a huge impact on interactivity because the browser layout and rendering engine won't have to move hundreds or thousands of elements on every position change but only an image element.

GenExp maintains an in-memory cache on the client side. The size of this cache has not been arbitrarily limited but thanks to the feature compression scheme applied on the server side, data is only received and so stored at the required resolution. Furthermore, cache management code will maintain the cache at manageable levels by removing unused data, with a process memory footprint for typical usage between tens of megabytes and a few hundreds of megabytes.

Two Javascript libraries are used in GenExp. The Prototype library (http://www.prototypejs.org/) extends the capabilities of the basic data types and hides cross-browser issues. In particular, Prototype's custom events implementation has been used to make the application event-driven. ExtJS (http://www.extjs.com/) has been used to create and manage the graphical user interface.

The same origin policy severely limits the communications of web applications to only the server it comes from. Since GenExp needs to get data from multiple distributed sources, it sends all its requests to a proxy in its own server, which in turn fetches the data from the DAS sources.

For some lightweight DAS queries (such as getting the metadata of a newly added source) that due to their minimal size would not

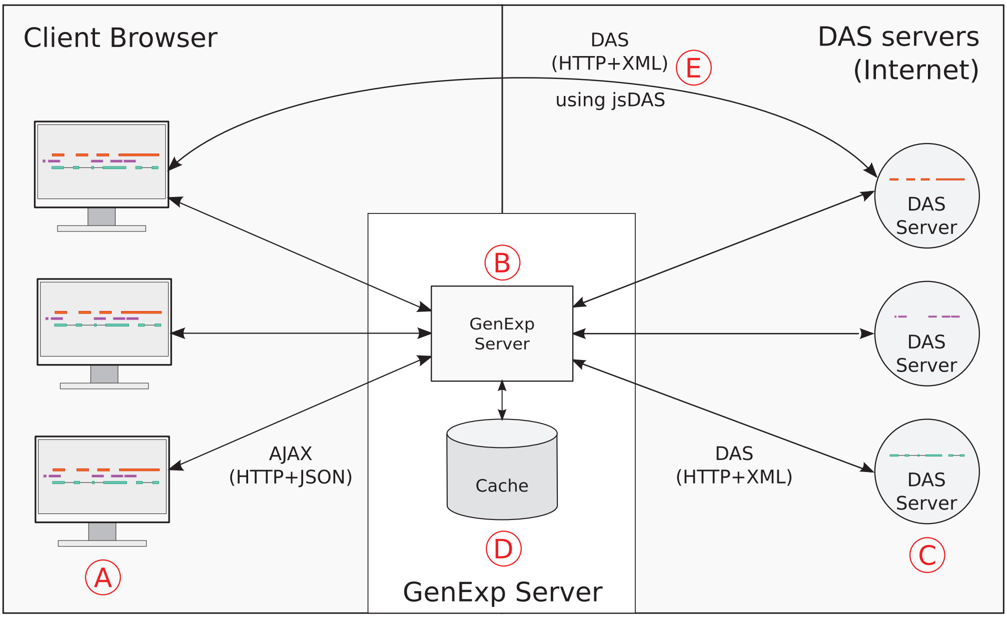


Figure 2. GenExp general architecture. The Javascript GenExp client (A) talks to the Perl GenExp server (B) using AJAX. The server asks the DAS servers (C) the required data, pre-processes it, stores it in the cache (D) and send it back to the clients to visualize it. When sending very lightweight queries, clients can directly connect to DAS servers using the jsDAS library (E).
doi:10.1371/journal.pone.0021270.g002

benefit from the the caching and compression performed by the GenExp server, the jsDAS library (http://code.google.com/p/jsdas) is used. jsDAS is a lightweight Javascript DAS client library and thanks to its Cross-Origin Resource Sharing (CORS) support, is able to connect to many DAS sources without the need of a server proxy. Since jsDAS does not implement any caching nor compression scheme, it would not be feasible to use it to retrieve feature data.

The server. The server subsystem is essentially a proxy and has three important additional capabilities: translation, preprocessing and caching. It has been written in Perl as a CGI script and uses the Bio::Das::Lite (http://search.cpan.org/dist/Bio-Das-Lite/) package to handle the DAS specific communication.

DAS data transport format is XML and while it is expressive, widely supported and human readable, it tends to be very verbose and somewhat clumsy to parse in Javascript. The JavaScript Object Notation (JSON) (http://json.org/), on the other hand, is simple, light and compact and native to Javascript and so is the data format used in the communication between GenExp client and server subsystems. The translation from DAS XML to GenExp JSON is performed by the server.

One of the key steps in providing zoom-dependent level of detail on the genome representation, is the server-side on-the-fly preprocessing. The server retrieves the data from the sources and creates different versions for different zoom levels, deciding on the required granularity.

The server has also caching capabilities and stores the preprocessed data for a short time, reducing the server and DAS sources load and speeding up responses. Since the caching times are short, updates on the data served by the DAS sources are propagated to the clients.

Availability and Future Directions

GenExp is an open source project hosted at http://code.google.com/p/genexp/ and under the GPL license. An instance of GenExp is running at http://gralggen.lsi.upc.edu/recerca/genexp and can be freely accessed with no restrictions.

In the near future, it is planned to further optimize GenExp so it can reliably work with denser data sources. It will also offer more customization options support for more complex renderings. Filtering, export and analysis capabilities are also planned.

Acknowledgments

The authors would like to thank Andrew M. Jenkinson and Rafael C. Jimenez for their helpful discussions and ideas. In addition, the authors would also like to thank the reviewers for their helpful suggestions.

Author Contributions

Conceived and designed the experiments: BGM XMP. Wrote the paper: BGM. Implemented the system: BGM.

References

1. Jenkinson AM, Albrecht M, Birney E, Blankenburg H, Down T, et al. (2008) Integrating biological data the Distributed Annotation System. BMC Bioinformatics 9: S3.
2. Dowell RD, Jokerst RM, Day A, Eddy SR, Stein L (2001) The distributed annotation system. BMC Bioinformatics 2: 7.
3. Hubbard TJP, Aken BL, Ayling S, Ballester B, Beal K, et al. (2009) Ensembl 2009. Nucleic Acids Research 37: D690–697.
4. Prlić A, Down TA, Kulesha E, Finn RD, Kähäri A, et al. (2007) Integrating sequence and structural biology with DAS. BMC Bioinformatics 8: 333.
5. Prlić A, Down TA, Hubbard TJP (2005) Adding some SPICE to DAS. Bioinformatics 21 Suppl 2: ii40–41.
6. Messina DN, Sonnhammer ELL (2009) DASher: a stand-alone protein sequence client for DAS, the Distributed Annotation System. Bioinformatics 25: 1333–1334.
7. Waterhouse AM, Procter JB, Martin DMA, Clamp M, Barton GJ (2009) Jalview Version 2–a multiple sequence alignment editor and analysis workbench. Bioinformatics 25: 1189–1191.
8. Macías JR, Jiménez-Lozano N, Carazo JM (2007) Integrating electron microscopy information into existing Distributed Annotation Systems. Journal of Structural Biology 158: 205–213.
9. Cases I, Pisano DG, Andres E, Carro A, Fernández JM, et al. (2007) CARGO: a web portal to integrate customized biological information. Nucleic acids research 35: W16–20.
10. Jimenez RC, Quinn AF, Garcia A, Labarga A, O'Neill K, et al. (2008) Dasty2, an Ajax protein DAS client. Bioinformatics 24: 2119–2121.
11. Nicol JW, Helt GA, Blanchard SG, Jr., Raja A, Loraine AE (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. Bioinformatics 25: 2730–2731.
12. Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, et al. (2011) Integrative genomics viewer. Nature Biotechnology 29: 24–26.
13. Kuhn RM, Karolchik D, Zweig AS, Wang T, Smith KE, et al. (2009) The UCSC Genome Browser Database: update 2009. Nucleic Acids Research 37: D755–D761.
14. Stein LD, Mungall C, Shu S, Caudy M, Mangone M, et al. (2002) The generic genome browser: a building block for a model organism system database. Genome Research 12: 1599–1610.
15. Gel Moreno B, Jenkinson AM, Jimenez RC, Messeguer Peypoch X, Hermjakob H (2011) easyDAS: Automatic creation of DAS servers. BMC bioinformatics 12: 23.

Chapter 7

Conclusions

With this thesis we wanted to tackle two problems regarding how researchers are able to share and disseminate their data and results and how they can use such data and visualize it in a useful way.

For the first problem, that of the sharing and dissemination of biological data, we have developed a solution to help biomedical researchers share their data through DAS, the Distributed Annotation System. easyDAS, the developed platform, offers the users a web-based interface to automatically create DAS sources with their data without any of the hassles associated with installing and managing DAS servers and even without any knowledge about DAS. With it, it is possible to create a DAS source with just a few clicks and automatically attach it to the Ensembl web browser to explore it in conjunction with other data and genomic annotations. easyDAS is open source software and is freely available. An instance of it is running at the IMPPC at http://gattaca.imppc.org/groups/eslab/easyDAS/.

Our approach to the second problem was to develop a technological prototype to test the feasibility of building a highly interactive genome browser with the web technologies available when the project started. The result was positive and a prototype web-based genome browser, GenExp, was built. It offers a highly interactive interface with direct manipulation of the genome representation. It also proved that building a usable genome browser with client-side data rendering was possible and that the use of a bitmap-based approach has some advantages over the use of object-based approaches such as SVG -although these advantages have been reduced thanks to the advances in web browser performance-. GenExp was also used as a test-bed for new user interface ideas for genomic browsers, such as multiple simultaneous viewers. GenExp is open source software and freely available and can be tested at http://gralggen.lsi.upc.edu/recerca/genexp/.

In addition, we developed jsDAS, a DAS client library implemented in javascript. With jsDAS is is possible to access DAS source right from the browser environment and greatly facilitates the development of web applications using DAS data sources.

We also participated in the development of two additional tools: Dasty3, the latest version of the EBI's protein annotation browser and myKaryoview, a genome browser aimed at users of direct to consumer genetic tests and other bioinformatics tinkerers.

This work led to the publication of four papers in peer-reviewed journals and presentations and demonstrations in a number of international conferences.

List of Acronims

aCGH: Array-based comparative genomic hibridization

API: Access Programming Interface

CORS: Cross-Origin Resource

DNA: Deoxyribonucleic acid

DOM: Document Object Model

DTC: Direct To Consumer

ENA: European Nucleotide Archive

FTP: File Transfer Protocol

GEO: Gene Expression Omnibus

GFF: General Feature Format

GRC: Genome Reference Consortium

GUI: Graphical User Interface

HTTP: Hypertext Transfer Protocol

HTML5: Hypertext Markup Language version 5.

MIAME: Minimum Information About a Microarray Experiment

miRNA: microRNA

MPNST: Malignant Peripheral Nerve Sheath Tumor

NGS: Next Generation Sequencing

NSCLC: Non-Small-Sell Lung Carcinoma

OLS: Ontology Lookup Service

REST: Respresentational State Transfer

RNA: Ribonucleic acid

SCNA: Somatic Copy Number Alteration

SNP: Single Nucleotide Polimorphism

SNP-array: SNP array

SOP: Same Origin Policy

SRA: Sequence Read Archive

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

UTR: Untranslated Region

XML: eXtensible Markup Language

XSLT: eXtensible Stylesheet Language Transformations

List of Figures

Bibliography

1. Oliphant, A., Barker, D. L., Stuelpnagel, J. R. & Chee, M. S. BeadArray technology: enabling an accurate, cost-effective approach to high-throughput genotyping. *BioTechniques* **Suppl**, 56–8, 60–1 (2002).

2. Lander, E. S. *et al.* Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).

3. The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature* **467**, 1061–73 (2010).

4. The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* **455**, 1061–8 (2008).

5. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic acids research* **30**, 207–10 (2002).

6. Parkinson, H. *et al.* ArrayExpress update–an archive of microarray and high-throughput sequencing-based functional genomics experiments. *Nucleic acids research* **39**, D1002–4 (2011).

7. Miller, G. A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review* **63**, 81–97 (1956).

8. Bauer, M. I. & Johnson-Laird, P. HOW DIAGRAMS CAN IMPROVE REASONING. *Psychological Science* **4**, 372–378 (1993).

9. Hubbard, T. J. P. *et al.* Ensembl 2009. *Nucleic Acids Research* **37**, D690–697 (2009).

10. Kuhn, R. M. *et al.* The UCSC Genome Browser Database: update 2009. *Nucleic Acids Research* **37**, D755–D761 (2009).

11. Stein, L. D. *et al.* The generic genome browser: a building block for a model organism system database. *Genome research* **12**, 1599–610 (2002).

12. Stein, L., Sternberg, P., Durbin, R., Thierry-Mieg, J. & Spieth, J. WormBase: network access to the genome and biology of Caenorhabditis elegans. *Nucleic acids research* **29**, 82–6 (2001).

13. Hubbard, T. *et al.* The Ensembl genome database project. *Nucleic acids research* **30**, 38–41 (2002).

14. Prlić, A. *et al.* Integrating sequence and structural biology with DAS. *BMC bioinformatics* **8**, 333 (2007).

15. The distributed annotation system. URL http://www.biodas.org/.

16. Specification of the distributed annotation system 1.6. URL http://www.biodas.org/wiki/DAS1.6.

17. Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. thesis, UNIVERSITY OF CALIFORNIA, IRVINE (2000).

18. Apweiler, R. *et al.* UniProt: the Universal Protein knowledgebase. *Nucleic acids research* **32**, D115–9 (2004).

19. Gehlenborg, N. *et al.* Visualization of omics data for systems biology. *Nature methods* **7**, S56–68 (2010).

20. Gel Moreno, B., Jenkinson, A. M., Jimenez, R. C., Messeguer Peypoch, X. & Hermjakob, H. easyDAS: Automatic creation of DAS servers. *BMC bioinformatics* **12**, 23 (2011).

21. Côté, R. G., Jones, P., Martens, L., Apweiler, R. & Hermjakob, H. The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. *Nucleic acids research* **36**, W372–6 (2008).

22. Finn, R. D. *et al.* ProServer: a simple, extensible Perl DAS server. *Bioinformatics (Oxford, England)* **23**, 1568–70 (2007).

23. jQuery: write less, do more. URL http://jquery.com.

24. Free Software Foundation. GNU Lesser General Public License (2012). URL http://www.gnu.org/licenses/lgpl.html.

25. Wang, J., Kong, L., Gao, G. & Luo, J. A brief introduction to web-based genome browsers. *Briefings in bioinformatics* (2012).

26. Google Maps. URL http://maps.google.com.

27. The canvas element - HTML standard. URL http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html.

28. Gel Moreno, B. & Messeguer Peypoch, X. GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering. *PLoS ONE* **6**, e21270 (2011).

29. Gel, B. & Messeguer, X. Implementing an Interactive Web-Based DAS Client. In Corchado, J., De Paz, J., Rocha, M. & Fernández Riverola, F. (eds.) *2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008)*, vol. 49 of *Advances in Soft Computing*, 83–91 (Springer Berlin / Heidelberg, 2009). URL http://dx.doi.org/10.1007/978-3-540-85861-4_11.

30. Prototype js: A foundation for ambitious web applications. URL http://prototypejs.org/.

31. Sencha ext js javascript framework for rich desktop apps. URL http://www.sencha.com/products/extjs.

32. GNU General Public License. URL http://www.gnu.org/copyleft/gpl.html.

33. Pettett, R. Bio::Das::Lite. URL http://search.cpan.org/~rpettett/Bio-Das-Lite-2.11/lib/Bio/Das/Lite.pm.

34. JDAS. URL http://code.google.com/p/jdas/.

35. Dasobert DAS client library. URL http://www.spice-3d.org/dasobert/index.jsp.

36. Gel, B. & Villaveces, J. M. jsDAS: Javascript client library for the Distributed Annotation System. URL http://code.google.com/p/jsdas/.

37. Navarro, A. *et al.* MicroRNA expression profiling in classic Hodgkin lymphoma. *Blood* **111**, 2825–2832 (2008).

38. Gallardo, E. *et al.* miR-34a as a prognostic marker of relapse in surgically resected non-small-cell lung cancer. *Carcinogenesis* **30**, 1903–1909 (2009).

39. Garcia-Linares, C. *et al.* Applying microsatellite multiplex PCR analysis (MMPA) for determining allele copy-number status and percentage of normal cells within tumors. *PloS one* **7**, e42682 (2012).

40. Rahrmann, E. P. *et al.* Forward genetic screen for malignant peripheral nerve sheath tumor formation identifies new genes and pathways driving tumorigenesis. *Nature genetics* 1–13 (2013).

41. on models for biomedical research, C. *Models for biomedical research: A new perspective* (National Academy Press, 1985).

42. Jenkinson, A. M. *et al.* Integrating biological data – the Distributed Annotation System. *BMC Bioinformatics* **9**, S3 (2008).

43. Gene Expression Omnibus. URL http://www.ncbi.nlm.nih.gov/geo/.

44. Frequently Asked Questions - GEO - NCBI. URL http://www.ncbi.nlm.nih.gov/geo/info/faq.htm\l#kinds.

45. Brazma, A. *et al.* ArrayExpress–a public repository for microarray gene expression data at the EBI. *Nucleic acids research* **31**, 68–71 (2003).

46. Brazma, A. *et al.* Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature genetics* **29**, 365–71 (2001).

47. Edgar, R. & Barrett, T. NCBI GEO standards and services for microarray data. *Nature biotechnology* **24**, 1471–2 (2006).

48. European Nucleotide Archive. URL http://www.ebi.ac.uk/ena/.

49. Genome Reference Consortium. URL www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/.

50. Dowell, R. D., Jokerst, R. M., Day, A., Eddy, S. R. & Stein, L. The Distributed Annotation System. *BMC Bioinformatics* **2** (2001).

51. Stein, L. D. & Thierry-mieg, J. Sequence and Other ACEDB Databases Scriptable Access to the Caenorhabditis elegans Genome Sequence and Other ACEDB Databases. *Genome Research* 1308–1315 (1998).

52. Finn, R. *et al.* eFamily: Bridging Sequence and Structure. In *Proceedings, UK e-Science, All Hands Meeting*, 1, 4–7 (2004). URL http://www.allhands.org.uk/2004/proceedings/papers/211.pdf.

53. Olason, P. I. Integrating protein annotation resources through the Distributed Annotation System. *Nucleic acids research* **33**, W468–70 (2005).

54. Thornton, J. Annotations for all by all - the BioSapiens network. *Genome biology* **10**, 401 (2009).

55. Reeves, G. a. & Thornton, J. M. Integrating biological data through the genome. *Human molecular genetics* **15 Spec No**, R81–7 (2006).

56. The DAS Registry. URL http://www.dasregistry.org.

57. Prlić, A., Down, T. A. & Hubbard, T. J. P. Adding some SPICE to DAS. *Bioinformatics* **21 Suppl 2**, ii40–41 (2005).

58. Eilbeck, K. *et al.* The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biology* **6**, R44 (2005).

59. Reeves, G. A. *et al.* The Protein Feature Ontology: a tool for the unification of protein feature annotations. *Bioinformatics (Oxford, England)* **24**, 2767–2772 (2008).

60. The Evidence Ontology. URL http://www.evidenceontology.org/.

61. Salazar, G. A. *et al.* MyDas, an extensible Java DAS server. *PloS one* **7**, e44180 (2012).

62. Visualizing Biological Data - Nature Methods. URL http://www.nature.com/nmeth/journal/v7/n3s/index.html.

63. Visualizing Biological Data - VIZBI. URL http://vizbi.org/.

64. NCBI Resource Coordinators. Database resources of the National Center for Biotechnology Information. *Nucleic acids research* **42**, D7–17 (2014).

65. Hubbard, T. J. P. *et al.* Ensembl 2009. *Nucleic Acids Research* **37**, D690–D697 (2009).

66. Yin, T., Cook, D. & Lawrence, M. ggbio: an R package for extending the grammar of graphics for genomic data. *Genome Biology* **13**, R77 (2012).

67. Durinck, S., Bullard, J., Spellman, P. T. & Dudoit, S. GenomeGraphs: integrated genomic data visualization with R. *BMC bioinformatics* **10**, 2 (2009).

68. Kin, T. & Ono, Y. Idiographica: a general-purpose web application to build idiograms on-demand for human, mouse and rat. *Bioinformatics (Oxford, England)* **23**, 2945–6 (2007).

69. Krzywinski, M. *et al.* Circos: an information aesthetic for comparative genomics. *Genome research* **19**, 1639–45 (2009).

70. Jones, L., Moszer, I. & Cole, S. T. Leproma: a Mycobacterium leprae genome browser. *Leprosy review* **72**, 470–7 (2001).

71. Stothard, P. & Wishart, D. S. Circular genome visualization and exploration using CGView. *Bioinformatics (Oxford, England)* **21**, 537–9 (2005).

72. Arakawa, K. *et al.* Genome Projector: zoomable genome map with multiple views. *BMC bioinformatics* **10**, 31 (2009).

73. Arakawa, K., Kido, N., Oshita, K. & Tomita, M. G-language genome analysis environment with REST and SOAP web service interfaces. *Nucleic acids research* **38**, 700–705 (2010).

74. Loraine, A. E. & Helt, G. a. Visualizing the genome: techniques for presenting human genome data and annotations. *BMC bioinformatics* **3**, 19 (2002).

75. Helt, G. A., Lewis, S., Loraine, A. E. & Rubin, G. M. BioViews: Java-based tools for genomic data visualization. *Genome research* **8**, 291–305 (1998).

76. Heumann, K., Harris, C. & Mewes, H. W. A top-down approach to whole genome visualization. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology* **4**, 98–108 (1996).

77. Helt, G. a. *et al.* Genoviz Software Development Kit: Java tool kit for building genomics visualization applications. *BMC bioinformatics* **10**, 266 (2009).

78. Nicol, J. W., Helt, G. A., Blanchard Jr., S. G., Raja, A. & Loraine, A. E. The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics* **25**, 2730–2731 (2009).

79. Thorvaldsdóttir, H., Robinson, J. T. & Mesirov, J. P. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics* (2012).

80. Lewis, S. E. *et al.* Apollo: a sequence annotation editor. *Genome biology* **3**, RESEARCH0082 (2002).

81. Misra, S. & Harris, N. Using Apollo to browse and edit genome annotations. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]* **Chapter 9**, Unit 9.5 (2006).

82. Engels, R. *et al.* Combo: a whole genome comparative browser. *Bioinformatics (Oxford, England)* **22**, 1782–3 (2006).

83. Abeel, T., Van Parys, T., Saeys, Y., Galagan, J. & Van de Peer, Y. GenomeView: a next-generation genome browser. *Nucleic acids research* **40**, e12 (2012).

84. Cline, M. & Kent, W. Understanding genome browsing. *Nature biotechnology* **27**, 153–155 (2009).

85. Nielsen, C. B., Cantor, M., Dubchak, I., Gordon, D. & Wang, T. Visualizing genomes: techniques and challenges. *Nature methods* **7**, S5–S15 (2010).

86. Lacroix, T., Loux, V., Gendrault, A., Gibrat, J.-F. & Chiapello, H. CompaGB: An open framework for genome browsers comparison. *BMC research notes* **4**, 133 (2011).

87. Kent, W. J. *et al.* The Human Genome Browser at UCSC. *Genome Research* **12**, 996–1006 (2002).

88. Skinner, M. E., Uzilov, A. V., Stein, L. D., Mungall, C. J. & Holmes, I. H. JBrowse: a next-generation genome browser. *Genome research* **19**, 1630–8 (2009).

89. Jimenez, R. C. *et al.* myKaryoView: A Light-Weight Client for Visualization of Genomic Data. *PLoS ONE* **6**, e26345 (2011).

90. Down, T. a., Piipari, M. & Hubbard, T. J. P. Dalliance: interactive genome viewing on the web. *Bioinformatics (Oxford, England)* **27**, 889–90 (2011).

91. Saito, T. L. *et al.* UTGB toolkit for personalized genome browsers. *Bioinformatics (Oxford, England)* **25**, 1856–61 (2009).

92. Miller, C. a., Anthony, J., Meyer, M. M. & Marth, G. Scribl: an HTML5 Canvas-based graphics library for visualizing genomic data over the web. *Bioinformatics (Oxford, England)* **29**, 381–3 (2013).

93. Genoverse - interactive HTML5 genome browser. URL http://www.genoverse.org/.

94. GenomeMaps. URL http://www.genomemaps.org/.

95. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic acids research* **28**, 235–42 (2000).

Part III

Appendices

Basic DAS Commands

This section contains a description of the most common parameters and then the list of the available commands along with a brief description of them and its specific parameters. The complete definition and specification of each command is available on the DAS 1.6 specification.

.0.1 Commands at Server level

The idea of server level commands is to retrieve metadata about the server and the data it serves. Only one command is available right now at the server level and is used to retrieve a list of the available data sources.

Sources

The `sources` command is the only command issued at server level. It is used to retrieve an XML document with a list of the data sources available on the server and some metadata about them. The `sources` command is required to be implemented by any DAS server, either a Reference Server or an Annotation Server.

Metadata about a source includes the coordinate system the data it serves is referenced to, the email of the person responsible of that source, its capabilities, name and URL.

It is possible to filter the sources to be listed by various criteria, mostly referring to the coordinate system.

Parameters

`sources` has no required parameters. All the available parameters specify filtering options to the retrieved list of sources. It is possible to filter by coordinate system parameters (`type` -"Chromosome", "Protein Sequence"-, `organism`, ...), `capability` (only the sources offering the `sequence` command, or those reporting their feature types) or `label` (to select only those with a given label assigned). It is possible to combine any of those filters to better specify the query.

Query Example

As an example, a list of the sources served from the uniprot DAS service with the sequence command implemented and serving annotations based on a coordinate system with *UniProt* as authority, the command should be:

http://www.ebi.ac.uk/das-srv/uniprot/das/sources?authority=UniProt\&capability=sequence

Response Document

The response document is a sources XML document. This document contains a list of SOURCE elements and for each of them, its information: URI, description, maintainer... and a list with all its enabled capabilities.

Response Example

Sending the request example above would return a document like that.

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet href="/das-srv/uniprot/xslt/sources_uniprot.xsl" type="text/xsl"?>
<SOURCES>
  <SOURCE uri="uniprot" doc_href="http://www.ebi.uniprot.org/index.shtml" title="UniProt" description="UniProt (Uni-
versal Protein Resource) is the world's most comprehensive catalog of information on proteins. It is a cen-
tral repository of protein sequence and function created by joining the information contained in Swiss-
Prot, TrEMBL, and PIR.">
    <MAINTAINER email="{rantunes, ljgarcia} AT ebi.ac.uk" />
    <VERSION uri="uniprot" created="2011-04-05">
      <COORDINATES uri="http://www.dasregistry.org/dasregistry/coordsys/CS_DS6" source="Protein Sequence" author-
ity="UniProt" test_range="O35502">UniProt,Protein Sequence</COORDINATES>
      <CAPABILITY type="das1:sources" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot" />
      <CAPABILITY type="das1:entry_points" query_uri="http://www.ebi.ac.uk/das-
srv/uniprot/das/uniprot/entry_points" />
      <CAPABILITY type="das1:sequence" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/sequence" />
      <CAPABILITY type="das1:types" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/types" />
      <CAPABILITY type="das1:features" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/features" />
      <CAPABILITY type="das1:stylesheet" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/stylesheet" />
      <CAPABILITY type="das1:error-segment" />
      <CAPABILITY type="das1:cors" />
    </VERSION>
  </SOURCE>
</SOURCES>
```

.0.2 Commands at Source level

The commands below are issued against a specific data source as opposed to the sources command, issued against the server itself. Most source level commands are used to retrieve different kinds of data, but some of them retrieve metadata about the data contained on the source.

Sequence

One of the two required commands for Reference Servers, the sequence command retrieves the actual sequence on the specified segment. The returned elements will usually be nucleotides for genomic sequences and aminoacids for protein sequences, although others elements might be possible for other sequence types.

Parameters

`sequence` has only one parameter, `segment`, and is required. It specifies the entry point and range whose sequence is required. It is possible to specify more than one segment on the same request.

Query Example

As an example, to retrieve the sequence for the first coding exon of the *BRCA2* gene, in the 30Mb region of chromosome 13, from the Ensembl! GRCH37 reference server, this command should be issued:

http://www.ensembl.org/das/Homo_sapiens.GRCh37.reference/sequence?segment=13:32890598, 32890664

Response Document

Response to the sequence command is a very simple XML document. It contains a single `SEQUENCE` element for each requested segment and the sequence itself as the content of that element.

Response Example

Sending the example request above would produce the following XML document:

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="/das/das.xsl"?>
<!DOCTYPE DASSEQUENCE SYSTEM "http://www.biodas.org/dtd/dassequence.dtd">
<DASSEQUENCE>
<SEQUENCE id="13" start="32890598" stop="32890664" version="1.0">
atgcctattggatccaaagagaggccaacatttttttgaaatttttaagacacgctgcaac
aaagcag
</SEQUENCE>
</DASSEQUENCE>
```

Entry Points

The `entry_points` command retrieves a list of the available reference sequences known by the data source. The identifiers returned by this command may be used to create valid segment specifications.

For each entry point, the `start` (usually 1) and `stop` (usually the length of the sequence) are included, as well as its `version` and `type`.

`entry_points` is a required command for a reference server and not required for annotation servers, where it's usually not implemented.

Parameters

The only parameter available for entry_points is rows, a pagination parameter. It is used for sources with information about many sequences (for example a protein sequence data source, where each protein is a reference sequence *per se*) to limit the number of entry points in the response. The format is *first-last*.

Query Example

As an example, this query would return 5 entry points (positions 10 to 15 on the complete entry points list) from the UniProt reference server, at the EBI.

http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/entry_points?rows=10-15

Response Document

The response is a DASEP XML document with a list of SEGMENT elements with the information about each sequence.

Response Example

Issuing the request above, this is whet would be received:

```
<?xml version="1.0" standalone="no"?>
<DASEP>
  <ENTRY_POINTS href="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/entry_points?rows=10-
15" version="2011.04" total="15082690" start="10" end="15">
    <SEGMENT id="A0A009" start="1" stop="530" ver-
sion="695874738beae14a6270a2c4a26afb8a" type="TrEMBL">A0A009_9ACTO</SEGMENT>
    <SEGMENT id="A0A010" start="1" stop="260" ver-
sion="d0286a9a93bc205cdfe180555818e8e8" type="TrEMBL">A0A010_9ACTO</SEGMENT>
    <SEGMENT id="A0A011" start="1" stop="281" ver-
sion="3d5bf92a3c2b7542b8371fe9c600dde1" type="TrEMBL">A0A011_9ACTO</SEGMENT>
    <SEGMENT id="A0A012" start="1" stop="226" ver-
sion="77ce56df7b82751280247c3b99b7957e" type="TrEMBL">A0A012_9ACTO</SEGMENT>
    <SEGMENT id="A0A013" start="1" stop="223" ver-
sion="dbfb47a92163d3dd6897d3af4644b7af" type="TrEMBL">A0A013_9ACTO</SEGMENT>
    <SEGMENT id="A0A014" start="1" stop="312" ver-
sion="d2cf0b3e80dc992914b4ab087dc18a34" type="TrEMBL">A0A014_9ACTO</SEGMENT>
  </ENTRY_POINTS>
</DASEP>
```

Features

The features command is the main data retrieval command. It is used to fetch annotations from an annotation server. Although not required, this command is implemented in most of the available DAS servers.

Parameters

All features parameters but one are filters to indicate which of the available features on the source should be returned. There are two main ways to filter: by position or by identifier. While position based filtering is required for every source implementing `features`, identifier based filtering is an optional capability and as such, must be reported via the capabilities information.

The `segment` parameter is used to filter y position. Specifying a sequence and optionally a range on that segment, the source will return only the features overlapping (at least partially) that segment. It is possible to specify more than one segment to retrieve the features in any of them.

When specifying one or more identifiers via the `feature_id` parameter, the source will return any feature overlapping with the ones asked for. For example, if a source serves genes, transcripts and exons, asking for a gene using the `feature_id` parameter will return the gene itself along with its exons and transcripts.

Those two main filtering options may be complemented with additional filters such as `type` or `category`, to further filter the list of features to include only those of the specified type or category.

An additional parameter specific to the `features` command is `maxbins`. On sources supporting this optional capability, a binning strategy is applied to the feature list. The binning strategy is defined by the data source implementator.

Query Example

As an example, knowing that the *p* arm of chromosome 18 has about 17Mb in length, this query to the karyotype DAS source from Ensembl! will retrieve its karyotype banding.

http://www.ensembl.org/das/Homo_sapiens.GRCh37.karyotype/features?segment=18:1,17000000

Response Document

The returned document is an XML DASGFF document. It is basically a list of `SEGMENT` elements and for each of them a list of `FEATURE` elements. Each `FEATURE` element contains all the information about the feature it represents: id, type, position, orientation... as well as its links, notes, parents and parts.

Response Example

Executing the query above would result in a returned document similar to the following.

Types

The `types` command is used to get meta-information about the content of the data source. Specifically which types of features it contains and how many of them.

Although required for any source implementing the `features` command, not every annotation server implements it.

Parameters

`types` accepts a single parameter: the `segment` specification. When no segment is specified the query will return all the types known to the source. If a segment is set, the query will return the types of the features overlapping that segment.

Query Example

The types of the UniProt's annotations for BRCA2 protein can be obtained using the following query:

http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/types?segment=Q9H265

Response Document

The response is a DASTYPES XML document with a list of `SEGMENT`'s and a list of `TYPE`'s for each one. For each type, only an identifier is required. The *category* and *cvID* (Controlled Vocabulary IDentifier,the ontological identifier) are optional. The count of features of that type is also optional.

Response Example

The response to the query above would be:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASTYPES SYSTEM "http://www.biodas.org/dtd/dastypes.dtd">
<DASTYPES>
  <GFF version="1.0" href="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/types?segment=Q9H265">
    <SEGMENT id="Q9H265" start="1" stop="58" version="e5f7f140a72d3f555c833992c63e910c">
      <TYPE id="SO:0001084" cvId="SO:0001084" category="inferred from electronic annotation (ECO:00000067)">2</TYPE>
      <TYPE id="BS:01034" cvId="BS:01034" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:00138" cvId="BS:00138" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:01040" cvId="BS:01040" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
      <TYPE id="BS:01003" cvId="BS:01003" category="inferred from electronic annotation (ECO:00000067)">1</TYPE>
    </SEGMENT>
  </GFF>
</DASTYPES>
```

Structure

`structure` is a command specific for those sources working with proteomics data. The command returns a complete description of the 3D structure of a protein or complex, acting as a reference server for 3D structures. It can return different models for the same protein and additional properties tied to the structure,

Parameters

query is a required parameter specifying the identifier of the protein. Two addition parameters, chain and model are used to filter the response.

Query Example

To get the structure of the BRCA2/PALB2 complex, one could use Sanger's structure server and give the identifier of the complex on the Protein Data Bank (PDB[95]) (http://www.pdb.org):

http://das.sanger.ac.uk/das/structure/structure?query=3EU7

Response Document

The response is an DASSTRUCTURE XML document with information about the object and its identification, a list of CHAIN's, with lists of GROUP's containing lists of ATOM's, everything extended with its additional parameters and meta-information so the complete 3D structure might be reconstructed.

Response Example

The response to the query above is a long document. An excerpt of it follows:

```
<?xml version='1.0' standalone='no' ?>
<dasstructure xmlns="http://www.efamily.org.uk/xml/das/2004/06/17/dasstructure.xsd"
  xmlns:data="http://www.efamily.org.uk/xml/data/2004/06/17/dataTypes.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.efamily.org.uk/xml/das/2004/06/17/dasstructure.xsd
  http://www.efamily.org.uk/xml/das/2004/06/17/dasstructure.xsd">
  <object dbAccessionId="3EU7" intObjectId="3EU7" objectVersion="15-SEP-09"
    type="protein structure" dbSource="PDB" dbVersion="20070116" dbCoordSys="PDBresnum,Protein Structure" />
  <chain id="A" SwissprotId="null">
    <group name="ASN" type="amino" groupID="854">
      <atom atomID="1" atomName=" N  " x="19.711" y="6.731" z="7.514" />
      <atom atomID="2" atomName=" CA " x="20.526" y="6.376" z="8.679" />
      <atom atomID="3" atomName=" C  " x="20.255" y="7.283" z="9.889" />
      <atom atomID="4" atomName=" O  " x="19.373" y="8.165" z="9.852" />
      <atom atomID="5" atomName=" CB " x="22.034" y="6.31" z="8.342" />
    </group>
    <group name="LEU" type="amino" groupID="855">
      <atom atomID="6" atomName=" N  " x="21.034" y="7.041" z="10.944" />
      <atom atomID="7" atomName=" CA " x="20.78" y="7.585" z="12.277" />
      <atom atomID="8" atomName=" C  " x="21.626" y="8.809" z="12.585" />
      <atom atomID="9" atomName=" O  " x="22.854" y="8.722" z="12.602" />
      <atom atomID="10" atomName=" CB " x="21.07" y="6.512" z="13.341" />
      <atom atomID="11" atomName=" CG " x="19.976" y="5.452" z="13.534" />
      <atom atomID="12" atomName=" CD1" x="20.336" y="4.434" z="14.61" />
      <atom atomID="13" atomName=" CD2" x="18.677" y="6.141" z="13.869" />
    </group>
    (...)
    <group name="HOH" type="hetatm" groupID="1282">
      <atom atomID="2585" atomName=" O  " x="2.096" y="-25.056" z="9.39" />
    </group>
  </chain>
  (...)
  <connect atomSerial="2487" type="bond">
    <atomID atomID="2486" />
  </connect>
  <connect atomSerial="2488" type="bond">
    <atomID atomID="2486" />
    <atomID atomID="2489" />
  </connect>
  <connect atomSerial="2489" type="bond">
    <atomID atomID="2488" />
  </connect>
</dasstructure>
```

Stylesheet

`stylesheet` is the only command at source level retrieving only metadata. It is used to get the rendering recommendations from the source, the *stylesheet*.

Parameters

`stylesheet` has no parameters.

Query Example

To know the recommended rendering style for mutation data on Sanger's COSMIC data, one should use the following query:

http://das.sanger.ac.uk/das/cosmic_mutations_GRCh37/stylesheet

Response Document

The response is a DASSTYLE XML document with a list of `CATEGORY` elements with a list of `TYPE`'s inside. Each of those `TYPE`'s contains the description of how the type should be rendered, mainly a `GLYPH` and its parameters.

Response Example

The response to the query above would be this document, stating that *substitution*'s should be represented with a blue cross, *deletion*'s with a downward-pointing red triangle and *insertion*'s with an upward-pointing blue triangle. Any other feature of another type or with no type information should be rendered as a a blue box.

```
<!DOCTYPE DASSTYLE SYSTEM "http://www.biodas.org/dtd/dasstyle.dtd">
<DASSTYLE>
<STYLESHEET version="1.0">
  <CATEGORY id="default">
    <TYPE id="default">
      <GLYPH>
        <BOX>
          <FGCOLOR>blue</FGCOLOR>
          <BGCOLOR>blue</BGCOLOR>
        </BOX>
      </GLYPH>
    </TYPE>
    <TYPE id="substitution">
      <GLYPH>
        <CROSS>
          <FGCOLOR>blue</FGCOLOR>
          <BGCOLOR>blue</BGCOLOR>
        </CROSS>
      </GLYPH>
    </TYPE>
    <TYPE id="deletion">
      <GLYPH>
        <TRIANGLE>
          <DIRECTION>S</DIRECTION>
          <FGCOLOR>blue</FGCOLOR>
          <BGCOLOR>blue</BGCOLOR>
        </TRIANGLE>
      </GLYPH>
    </TYPE>
    <TYPE id="insertion">
      <GLYPH>
        <TRIANGLE>
          <DIRECTION>N</DIRECTION>
          <FGCOLOR>blue</FGCOLOR>
          <BGCOLOR>blue</BGCOLOR>
        </TRIANGLE>
      </GLYPH>
    </TYPE>
  </CATEGORY>
</STYLESHEET>
</DASSTYLE>
```

No command

Additionally to the commands above, sending a request to the source URL with no command or other arguments returns an XML document describing the source. The format is exactly the same as in the SOURCES command, but restricted to that single source.

Query Example

As an example, querying the UniProt data source with no command nor parameters as in

http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/

Would return a document such as that:

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet href="/das-srv/uniprot/xslt/sources_uniprot.xsl" type="text/xsl"?>
<SOURCES>
  <SOURCE uri="uniprot" doc_href="http://www.ebi.uniprot.org/index.shtml" title="UniProt" description="UniProt (Uni-
versal Protein Resource) is the world's most comprehensive catalog of information on proteins. It is a cen-
tral repository of protein sequence and function created by joining the information contained in Swiss-
Prot, TrEMBL, and PIR.">
    <MAINTAINER email="{rantunes, ljgarcia} AT ebi.ac.uk" />
    <VERSION uri="uniprot" created="2011-04-05">
      <COORDINATES uri="http://www.dasregistry.org/dasregistry/coordsys/CS_DS6" source="Protein Sequence" author-
ity="UniProt" test_range="O35502">UniProt,Protein Sequence</COORDINATES>
      <CAPABILITY type="das1:sources" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot" />
      <CAPABILITY type="das1:entry_points" query_uri="http://www.ebi.ac.uk/das-
srv/uniprot/das/uniprot/entry_points" />
      <CAPABILITY type="das1:sequence" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/sequence" />
      <CAPABILITY type="das1:types" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/types" />
      <CAPABILITY type="das1:features" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/features" />
      <CAPABILITY type="das1:stylesheet" query_uri="http://www.ebi.ac.uk/das-srv/uniprot/das/uniprot/stylesheet" />
      <CAPABILITY type="das1:error-segment" />
      <CAPABILITY type="das1:cors" />
    </VERSION>
  </SOURCE>
</SOURCES>
```

Additional publications related to the thesis

Two additional papers highly related to the thesis content have been published. The first presents myKaryoview, an integrative browser specialized in data from direct-to-consumer (DTC) genetic testing providers and aimed at bioinformatics tinkerers has been published in PLOS One. The second paper is devoted to Dasty3, the third incarnation of EBI's protein browser, and was published in Bioinformatics.

OPEN ACCESS Freely available online PLoS one

myKaryoView: A Light-Weight Client for Visualization of Genomic Data

Rafael C. Jimenez[1], Gustavo A. Salazar[2], Bernat Gel[3], Joaquin Dopazo[4,5], Nicola Mulder[2], Manuel Corpas[6]*

1 European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, United Kingdom, **2** Computational Biology Group, Department of Clinical Laboratory Sciences, IIDMM, University of Cape Town, Cape Town, South Africa, **3** Universitat Politècnica de Catalunya, Barcelona, Spain, **4** Centro de Investigación Príncipe Felipe, Valencia, Spain, **5** Functional Genomics Node (INB), CIPF, Valencia, Spain, **6** Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, United Kingdom

Abstract

The Distributed Annotation System (DAS) is a protocol for easy sharing and integration of biological annotations. In order to visualize feature annotations in a genomic context a client is required. Here we present myKaryoView, a simple light-weight DAS tool for visualization of genomic annotation. myKaryoView has been specifically configured to help analyse data derived from personal genomics, although it can also be used as a generic genome browser visualization. Several well-known data sources are provided to facilitate comparison of known genes and normal variation regions. The navigation experience is enhanced by simultaneous rendering of different levels of detail across chromosomes. A simple interface is provided to allow searches for any SNP, gene or chromosomal region. User-defined DAS data sources may also be added when querying the system. We demonstrate myKaryoView capabilities for adding user-defined sources with a set of genetic profiles of family-related individuals downloaded directly from 23andMe. myKaryoView is a web tool for visualization of genomic data specifically designed for direct-to-consumer genomic data that uses publicly available data distributed throughout the Internet. It does not require data to be held locally and it is capable of rendering any feature as long as it conforms to DAS specifications. Configuration and addition of sources to myKaryoView can be done through the interface. Here we show a proof of principle of myKaryoView's ability to display personal genomics data with 23andMe genome data sources. The tool is available at: http://mykaryoview.com.

Citation: Jimenez RC, Salazar GA, Gel B, Dopazo J, Mulder N, et al. (2011) myKaryoView: A Light-Weight Client for Visualization of Genomic Data. PLoS ONE 6(10): e26345. doi:10.1371/journal.pone.0026345

Editor: Francesc Palau, Instituto de Ciencia de Materiales de Madrid - Instituto de Biomedicina de Valencia, Spain

Received June 1, 2011; **Accepted** September 25, 2011; **Published** October 26, 2011

Funding: This work has been supported by ENFIN (contract number LSHG-CT-2005518254) and the Spanish Ministry of Science and Innovation (Subprograma de Acciones Integradas, HS2008-0010). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: mc10@sanger.ac.uk

Introduction

Advances in genome sequencing and screening technologies are producing genomic data at an unprecedented scale. Direct-to-consumer (DTC) genetic testing is also becoming relatively successful at attracting people who would like to have their genetic profile genotyped. Genetic profiles, however, provide little biological insight unless they are compared to other relevant data sources. In order to extract any meaning from genetic profiles containing single nucleotide polymorphisms (SNPs), copy number variations (CNVs) or specific genomic variants, raw genome data should be analysed in the context of other genes and annotations. A wealth of databases and annotation resources are available over the Internet with data that can help enrich results obtained in direct-to-consumer tests. Of particular relevance are resources such as HGNC [1] that gives unique and coherent nomenclature for genes, which can then be mapped to specific genome coordinates, the On-line Mendelian Inheritance in Man (OMIM) database [2] that characterises genes and syndromes involved in diseases inherited in a Mendelian fashion, the Database of Genomic Variants [3] that collects mostly CNVs reported in normal individuals, or the Database of Somatic Mutations in Cancer (COSMIC) [4]. These resources may prove useful when determining the potential genetic origin of specific traits in the person. In order to integrate genetic data together into a single interface it is usually necessary to locate the database or resource, download the data, write a specific parser for it and insert matched snippets of information into another database that can be queried against. In addition, this process of retrieval of data has to be periodically repeated to make sure that the retrieved information is up-to-date.

The Distributed Annotation System (DAS) is a widely used protocol for the exchange of biological information using XML [5]. DAS has the advantage of dramatically facilitating the process of integration of disparate biological annotations, providing one single mode of access via a RESTful web service. A DAS registry containing published DAS resources allows the browsing of relevant DAS sources. There are DAS sources for many different organisms and reference assemblies. By selecting a reference assembly, human variation data sources listed in the DAS registry can be easily integrated and mapped via a common coordinate reference system.

Genome visualization resources like Ensembl [6] rely heavily on data sources available via DAS. Ensembl is a very sophisticated system used for a great variety of species with many different kinds of data, allowing relatively complex integration and search queries. Ensembl is particularly useful for integration of many data sources, shown as tracks at different regions in the genome. Despite Ensembl's and other browsers' [7,8] ability to show different levels of annotation, every time coordinates are changed, the page needs to be refreshed to reload all the information with no simultaneous visualisation of different zoom levels. Furthermore, visualisation of karyotypes is either disallowed or too impractical to allow sufficient interactivity for the user when wanting to view information at the genome level. Therefore, we designed a flexible tool for visualisation of genomic DAS annotations that is complementary to existing genome browsers and yet specifically tailored to visualization of DTC genomic data. This tool, named *myKaryoView*, allows easy navigation between different levels of scale in three integrated 'views' that interact with each other. myKaryoView is thus a client that operates under a single interface and constitutes a "one stop shop" for simple and rapid queries, not requiring much knowledge of either bioinformatics tools or information technology in order to operate it. myKaryoView provides three views: karyotype, chromosome and zoom. Once the data is rendered, further requests are not needed to navigate freely within the region (unless zooming out). Each DAS data source is represented in different tracks with annotations that are clickable and provide further information about that particular feature and links to the relevant sources. Links back to Ensembl from any selected region allow automatic loading of the region in the Ensembl browser. A simple interface is provided to query myKaryoView by SNP id, gene name or genomic region as well as selection of user-defined DAS sources.

myKaryoView has been specifically configured to provide an integrated view of sources for analysis of DTC genomic data. But it can also be used for visualization of any genomic DAS data feature as long as it is in the same coordinate of reference. myKaryoView is able to provide a genome-wide perspective and easy navigation between different levels of detail with no need for refreshing the page, giving an overall enhanced visualization experience for genome analysis. myKaryoView's search interface and navigation widgets provide a variety of options that may help shed light on interpretation of genetic profiles and personal genomics data in general. myKaryoView does not require a lot of specialised knowledge in order to operate its simplest functions, although biological knowledge is required in order to appropriately understand the data sources integrated by default in the browser. Its human variation genome-focused configuration of views and sources of data make it appropriate for direct-to-consumer genome data analysis.

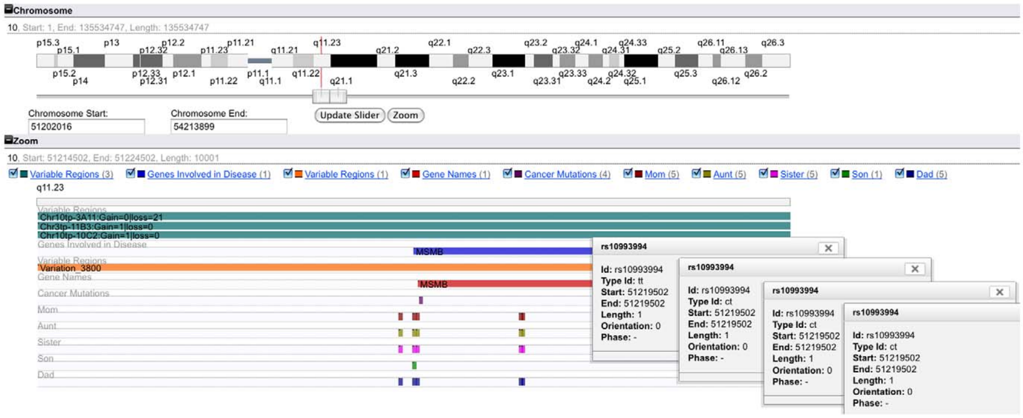
Results

Genome Navigation

Several widgets are provided for easy navigation and further exploration of retrieved annotation features. The chromosome view has a slider widget for zooming in or out of selected chromosomal regions. Moving the right and left slider to the desired region to be zoomed in and clicking "Zoom" refreshes the zoom view to this region. Any band in the chromosome view can also be clicked on and automatically displayed in the zoom view. Annotations are shown in different tracks with their respective legends at the top of the view. The legend shows the colour in which a particular annotation is displayed and their number in parentheses. Clicking on the legend, a popup window appears with a link "Display Annotations in Ensembl". This popup window also provides a link for viewing of the original data as it is retrieved via DAS. Any of the tracks can be selected/deselected by checking/unchecking the check box next to the legend. Any chromosome in the karyotype view, if clicked on, is automatically reloaded in the chromosome and zoom views. By clicking on any of the features, another popup window appears with more specific annotation about the feature and, where applicable, links to the original source. All of the data for any of the displayed sources can also be downloaded if the legend is clicked and the 'Show Original Source' link is selected. A new tab is thus opened with a formatted version of the raw data as retrieved from the server. This is especially useful if further analyses are sought with the visualised data, as this view allows easy copying and pasting of data as text, containing all annotations for a particular data source. Once the search query is performed there is little need to go back to the initial interface for navigation across any region in the genome.

We provide proof-of-principle results to show that a) it is possible to add user-defined data for visualisation of direct-to-consumer genome data and b) it can be used as a stand-alone genome browser for visualization of human genetic variation data.

a) Personal Genomics Family Case. A 23andMe customer ("son") wanted to find more information about the genetic causes leading to his increased risk of prostate cancer. An analysis of his genome profile yields a 28.1% risk of developing the cancer as opposed to the 17.8% average risk in males. This risk is calculated analysing the genotypes of 12 SNPs. The SNP marker rs10993994 shows the greatest risk among the 12 reported markers, a 1.3 times increased odds. This SNP is located in 10q11, near the *MSMB* gene and the effects of the identified allele (T) have been shown to decrease its expression levels [9], thus decreasing its cancer suppressor function. Having no history of prostate cancer in close relatives, 4 family members for the same customer were genotyped (mother, father, sister and aunt) and their data was made into DAS source in order to compare the pattern of inheritance between different family members for this SNP using myKaryoView. All these individual's genome profiles, were downloaded from 23andMe and a DAS source was created using easyDAS [10]. The resulting data sources were held privately in his newly created easyDAS account. The DAS URLs for the private sources were pasted into the myKaryoView interface and all other sources were selected to display in the zoom view. The 'rs10993994' SNP was typed in the query search box. Once results were returned, the zoom view was centred around the rs10993994 SNP±100 Kb of the SNP position (51219502) (Figure 1). Next, the pattern of inheritance for this SNP was elucidated by clicking on the SNP in the other members of the family analyzed. Clicking on the rs10993994 SNP feature in each family member track triggers a popup window with information about the SNP, among them the Type Id, which provides the actual genotype for that position. It was found that the father and the mother both have CT, with no risk associated with this variant, while the son inherited a T from each parent, producing the TT genotype with increased disease risk. The density of the number of analyzed SNPs is also clearly depicted by myKaryoView, showing that the son has fewer analyzed SNPs for this region than the other members of the family. This is due to the fact that a new SNP chip version was used with double the number of features for the other members of the family. Furthermore, it was noted that the start position of the MSMB gene is 51219559, only 57 bp after the SNP position. The Cancer Mutations track contains four reported mutations for MSMB (MSMB:ENST00000358559), indicative of the involvement of this gene in cancer, but all of them within the gene's exons. The overlap of four normal CNVs in the two Variable Region tracks show this is a hugely variable region. Through such close proximity of this SNP to the MSMB gene and the recorded



10, Start: 1, End: 135534747, Length: 135534747

Chromosome Start: 51202016

Chromosome End: 54213899

10, Start: 51214502, End: 51224502, Length: 10001

rs10993994

Id: rs10993994
Type Id: tt
Start: 51219502
End: 51219502
Length: 1
Orientation: 0
Phase: -

rs10993994

Id: rs10993994
Type Id: ct
Start: 51219502
End: 51219502
Length: 1
Orientation: 0
Phase: -

rs10993994

Id: rs10993994
Type Id: ct
Start: 51219502
End: 51219502
Length: 1
Orientation: 0
Phase: -

rs10993994

Id: rs10993994
Type Id: ct
Start: 51219502
End: 51219502
Length: 1
Orientation: 0
Phase: -

Figure 1. Searching myKaryoView by SNP. The rs10993994 SNP was identified in a 23andMe report to contribute most to the disease risk of prostate cancer in a customer, implicated by the MSMB gene. In addition to the customer, 4 additional members of his family uploaded their genotypes into myKaryoView for elucidation of how this particular SNP genotype was inherited. Popup windows for the SNP are shown in this order: son, mother, father and sister. The Type Id field shows their genotype for this position in their genome, TT, CT, CT, CT respectively. The aunt (not shown) has a CC genotype for this position.
doi:10.1371/journal.pone.0026345.g001

variation, findings are consistent with the observation that an allele variant found in the 23andMe analysis could potentially have an effect on the promoter region of MSMB, suggesting a causal variant at 10q11 that confers increased risk of prostate cancer. myKaryoView is thus able to provide a convenient visualization interface for easily navigating and querying any SNP the DTC customer might be interested in. Although it is not a substitute for the interpretation of the genomic data tested in an individual, it provides a contextualization and perhaps a solution to start exploring data sources that might help elucidate phenotypic associations. By integrating the genotypes of different family members, myKaryoView provides a convenient search interface for visualization of analyzed feature density and ascertaining patterns of inheritance in related individuals.

b) Human Genetic Variation Analysis Case. myKaryoView can also serve the purposes of a generic genome browser in addition to its application to personal genomics shown above. We demonstrate its generic capabilities in this second use case.

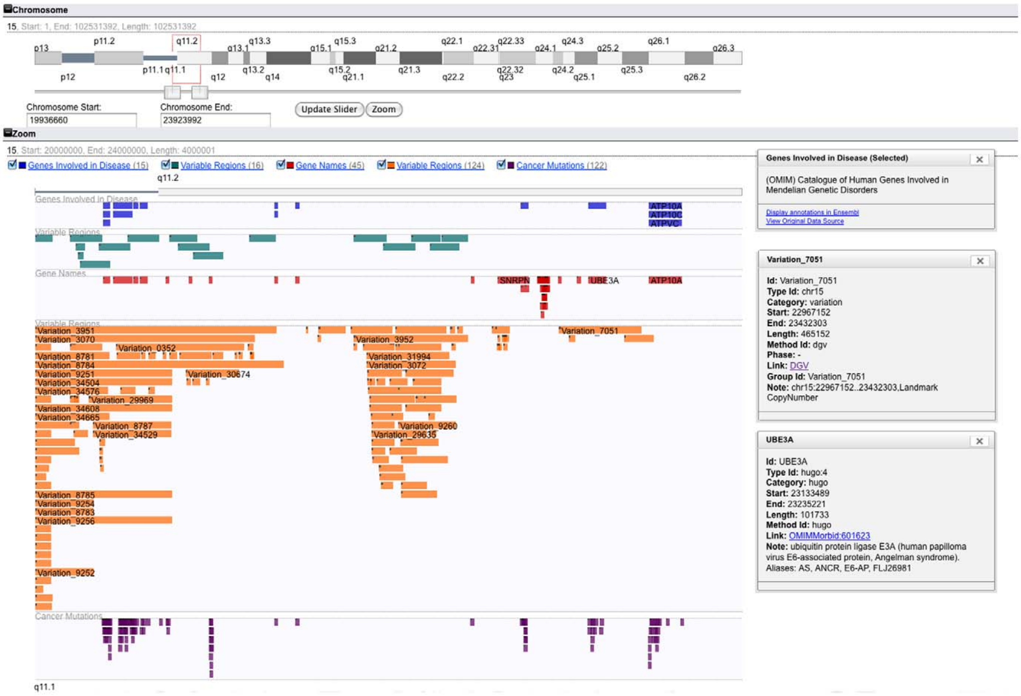
Deletion of the 15q11 region has been related to Prader-Willi and Angelman Syndromes [11,12,13]. This region roughly corresponds to the interval 20000000–24000000 in chromosome 15. According to the DECIPHER database [14], deletion of this region results in mental retardation and developmental delay. In order to explore genes likely to affect the patient phenotype, myKaryoView was utilised to look for genes involved in disease that overlapped this region and were reported to be involved in Mendelian disease inheritance, as well as regions of normal copy number variation. All available DAS sources were selected to display in the zoom view as tracks. In the input search box, the string '15:20000000,24000000' was entered.

Figure 2 shows results obtained from this search query. Fifteen genes involved in disease were retrieved from a total of forty-five genes in this region. Clicking on the 'Genes Involved in Disease' legend, a popup window appears with the option 'View Original Data Source', which triggers a new tab when clicked, with the original data source as obtained from the source (OMIM). myKaryoView results show greater density of 'normal' CNVs in gene desert regions. One notable exception to this pattern is Ubiquitin protein ligase E3A, a gene reported to produce an abnormal phenotype in ubiquitin-mediated protein degradation during brain development when this region is deleted [15]. This gene overlaps with Variation 7051 in DGV, a 0.4 Mb deletion reported by de Smith et al (2007) [16], using an Agilent 185 k CGH Array. We notice that a nearby cluster of genes of the SNORD family are included in Variation 7051 but are outside the myKaryoView graph, showing that genome coordinates for different sources may vary slightly. Moreover, due to array resolutions, it is possible that genes reported to be included in a normal CNV may, in reality, be misreported as such due to variable resolution of CNV detection methods.

Discussion

User Interaction Capabilities

A selection of human variation data sources is available by default in myKaryoview's website. These are DAS sources known to be relevant for genome variation analyses: HGNC Genes, OMIM, COSMIC, DGV and Redon clones [17]. In order to make them intelligible to lay users we have named them as: gene names, genes involved in disease, cancer mutations and variable regions. To make a request, at least one source has to be selected or uploaded. Input parameters are provided for users in the query interface for the following options: a) type of feature visualisation device (mark, track, line or chart), b) level of detail (views) in which to visualise the selected data source (karyotype, chromosome or zoom) and c) an input text box for specification of user-defined DAS sources. A URL with a valid DAS source address is required for user-defined DAS sources. Currently the interface is designed to allow one user-defined source, but this capability will be expanded in the near future to allow simultaneous visualization of several family member genomes. Existing genomic DAS sources available in the DAS registry are suitable for the input text box, as long as the source is built on the same reference assembly as the one used by myKaryoView (currently *Homo sapiens* NCBI36).



Chromosome

15. Start: 1, End: 102531392, Length: 102531392

Chromosome Start: 19936660

Chromosome End: 23923992

Zoom

15. Start: 20000000, End: 24000000, Length: 4000001

☑ Genes Involved in Disease (15) ☑ Variable Regions (16) ☑ Gene Names (45) ☑ Variable Regions (124) ☑ Cancer Mutations (122)

Genes Involved in Disease (Selected)

(OMIM) Catalogue of Human Genes Involved in Mendelian Genetic Disorders

Display annotations in Ensembl
View Original Data Source

Variation_7051

Id: Variation_7051
Type Id: chr15
Category: variation
Start: 22967152
End: 23432303
Length: 465152
Method Id: dgv
Phase: -
Link: DGV
Group Id: Variation_7051
Note: chr15:22967152..23432303,Landmark CopyNumber

UBE3A

Id: UBE3A
Type Id: hugo:4
Category: hugo
Start: 23133489
End: 23235221
Length: 101733
Method Id: hugo
Link: OMIMMorbid:601623
Note: ubiquitin protein ligase E3A (human papilloma virus E6-associated protein, Angelman syndrome). Aliases: AS, ANCR, E6-AP, FLJ26981

Figure 2. Chromosome Location Search. Searching 15:20000000,24000000 displays part of the 15q11 chromosomal band in zoom view. All listed DAS sources (gene names, genes involved in disease, cancer mutations and variable regions) were selected, choosing zoom and track visualisation options. The total number of features per track is shown next to the legend in parentheses. Forty five genes lie in selected intervals. Clicking on the 'Genes Involved in Disease' legend, a popup window appears providing links to the region in Ensembl and the original raw data that can be easily cut and pasted. The UBE3A gene bar is clicked and another popup window appears with links to further information. Variation 7051 reported by DGV is also clicked.
doi:10.1371/journal.pone.0026345.g002

Alternatively, user-defined DAS sources can be created. easyDAS is a tool that simplifies the process of DAS source creation and is able to convert genomic data from a variety of formats into a valid source which can be public or private. The created URL for the DAS source can then be added to the myKaryoView interface and visualised as any other data source.

The final requirement to run a query concerns entering the actual region or gene to be visualized. myKaryoView allows the search of any chromosomal location in the genome. A chromosome, and a start and end position separated by a colon and comma respectively (e.g. 1:2000000,3000000), constitutes the valid format for a chromosome location search. Searching can also be accomplished using valid HGNC gene names (also known as symbols). If searching by genes, the user should start by typing any character and selecting the appropriate gene name from a drop-down box that appears with suggestions. Once a gene is chosen from the list, one can hit 'Submit Query'. Results will appear as they are retrieved, with the zoom view centred on the start and end coordinates of the entered chromosomal location or the start and end position of the chosen gene.

Availability and Future Directions

The software can be freely used through the website http://mykaryoview.com. The source code is also available and can be downloaded as a web application under an Apache 2.0 license. We anticipate that direct-to-consumer genetic test customers would benefit from using myKaryoView by further exploring and discovering new insights not available in their test results. We envisage myKaryoView as a simple-to-use complementary resource to more sophisticated genome browsers.

Methods

Implementation and Design

myKaryoView is a light-weight client specifically designed to browse genomic data available via DAS with no data stored locally. Currently myKaryoView is configured to display *Homo sapiens* NCBI36 genome assembly data features, but it can also be used with alternative assemblies or even different organisms by simply modifying the configuration file in its source code, for which a manual is provided for advanced users. Although the NCBI36 assembly is not the most up-to-date human assembly, we chose this one to maintain compatibility with the raw bed file data provided by 23andMe, which currently is NCBI36. The source code is freely available for download under a CCA-SA license and authors welcome requests of support from potential collaborators. The fact that it is mainly written in cross-browser javascript code means that myKaryoView should work in every major browser and if installed externally it should work automatically in a web server directory. Currently we have tested myKaryoView in Internet Explorer 6 and, 7, Mozilla Firefox and Chrome. myKaryoView works well when the total number of features to be rendered in one go is in the order of several thousand. Although whole genome requests can, in principle, be handled by myKaryoView, if too many features are requested, this may overwhelm the computer's RAM capacity. Another limitation lies in the time it takes for data retrieval. Since all the data has to be retrieved from the Internet, slow Internet connections may negatively impact the user's experience. For most queries however, retrieval and rendering of data should be achieved in a few seconds.

Acknowledgments

We are grateful to Paul Bevan for commenting on the manuscript and helpful feedback.

Author Contributions

Conceived and designed the experiments: RCJ MC NM JD. Performed the experiments: RCJ MC. Analyzed the data: MC. Contributed reagents/materials/analysis tools: GAS BG. Wrote the paper: MC.

References

1. Bruford EA, Lush MJ, Wright MW, Sneddon TP, Povey S, et al. (2008) The HGNC Database in 2008: a resource for the human genome. Nucleic Acids Res 36: D445–448.
2. Hamosh A, Scott A, Amberger J, Bocchini C, McKusick V (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research 33: D514.
3. Iafrate A, Feuk L, Rivera M, Listewnik M, Donahoe P, et al. (2004) Detection of large-scale variation in the human genome. Nature genetics 36: 949–951.
4. Forbes S, Tang G, Bindal N, Bamford S, Dawson E, et al. (2009) COSMIC (the Catalogue of Somatic Mutations in Cancer): a resource to investigate acquired mutations in human cancer. Nucleic acids research.
5. Jenkinson AM, Albrecht M, Birney E, Blankenburg H, Down T, et al. (2008) Integrating biological data - the Distributed Annotation System. BMC Bioinformatics 9 Suppl 8: S3.
6. Flicek P, Amode M, Barrell D, Beal K, Brent S, et al. (2010) Ensembl 2011. Nucleic acids research.
7. Sanborn J, Benz S, Craft B, Szeto C, Kober K, et al. (2010) The UCSC cancer genomics browser: update 2011. Nucleic acids research.
8. Down TA, Piipari M, Hubbard TJP (2011) Dalliance: interactive genome viewing on the web. Bioinformatics 27: 889.
9. Chang B, Cramer S, Wiklund F, Isaacs S, Stevens V, et al. (2009) Fine mapping association study and functional analysis implicate a SNP in MSMB at 10q11 as a causal variant for prostate cancer risk. Human molecular genetics 18: 1368.
10. Gel B, Jenkinson A, Jimenez R, Peypoch XM, Hermjakob H (2011) easyDAS: Automatic creation of DAS servers. BMC Bioinformatics 12: 23.
11. Buiting K, Saitoh S, Gross S, Dittrich B, Schwartz S, et al. (1995) Inherited microdeletions in the Angelman and PraderñWilli syndromes define an imprinting centre on human chromosome 15. Nature genetics 9: 395–400.
12. Cassidy S, Driscoll D (2008) PraderñWilli syndrome. European Journal of Human Genetics 17: 3–13.
13. Williams C, Angelman H, Clayton Smith J, Driscoll D, Hendrickson J, et al. (1995) Angelman syndrome: consensus for diagnostic criteria. American Journal of Medical Genetics 56: 237–238.
14. Firth HV, Richards SM, Bevan AP, Clayton S, Corpas M, et al. (2009) DECIPHER: Database of Chromosomal Imbalance and Phenotype in Humans Using Ensembl Resources. Am J Hum Genet 84: 524–533.
15. Kishino T, Lalande M, Wagstaff J (1997) UBE3A/E6-AP mutations cause Angelman syndrome. Nature genetics 15: 70–73.
16. De Smith A, Tsalenko A, Sampas N, Scheffer A, Yamada N, et al. (2007) Array CGH analysis of copy number variation identifies 1284 new genes variant in healthy white males: implications for association studies of complex diseases. Human molecular genetics 16: 2783.
17. Redon R, Ishikawa S, Fitch K, Feuk L, Perry G, et al. (2006) Global variation in copy number in the human genome. Nature 444: 444–454.

BIOINFORMATICS APPLICATIONS NOTE

Vol. 27 no. 18 2011, pages 2616–2617
doi:10.1093/bioinformatics/btr433

Databases and ontologies

Advance Access publication July 28, 2011

Dasty3, a WEB framework for DAS

Jose M. Villaveces[1], Rafael C. Jimenez[1], Leyla J. Garcia[1], Gustavo A. Salazar[1,2], Bernat Gel[3], Nicola Mulder[2], Maria Martin[1], Alexander Garcia[4] and Henning Hermjakob[1,*]

[1]Protein and Nucleotide Data Group, EMBL Outstation European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK, [2]Computational Biology Group, Department of Clinical Laboratory Sciences, University of Cape Town, Cape Town, South Africa, [3]Software Department, UPC-BarcelonaTech, Barcelona, Spain and [4]Biomedical Informatics department, University of Arkansas for Medical Sciences, Little Rock, AR 72205, USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Dasty3 is a highly interactive and extensible Web-based framework. It provides a rich Application Programming Interface upon which it is possible to develop specialized clients capable of retrieving information from DAS sources as well as from data providers not using the DAS protocol. Dasty3 provides significant improvements on previous Web-based frameworks and is implemented using the 1.6 DAS specification.

Availability: Dasty3 is an open-source tool freely available at http://www.ebi.ac.uk/dasty/ under the terms of the GNU General public license. Source and documentation can be found at http://code.google.com/p/dasty/.

Contact: hhe@ebi.ac.uk

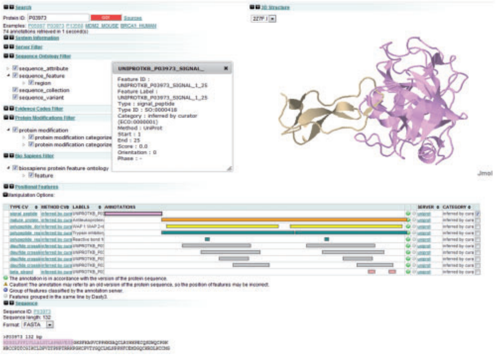
Received on April 8, 2011; revised on July 15, 2011; accepted on July 18, 2011

1 INTRODUCTION

The Distributed Annotation System (DAS) defines a communication protocol used to exchange annotations on genomic or protein sequences (Jenkinson *et al.*, 2008). Implicit in the conception of DAS is the notion of decentralization; annotations should not be provided by single centralized databases, but should instead be spread over multiple sites. DAS relies on client–server architecture; DAS servers manage data distribution by following a standard protocol, while manipulation and visualization are on the client side –thus facilitating the development of tailored tools addressing specific requirements. Several clients are available; for instance, SPICE (Prlic *et al.*, 2005) is a browser that displays protein sequences, structures and their corresponding annotations. Another client, Dasty2, provides a Web-based interface that facilitates visualizing and comparing protein sequence annotations from DAS servers (Jimenez *et al.*, 2008). These clients are addressing specific needs; however, none of them is providing a framework upon which tools can be built and integrated into a cohesive web environment.

2 RESULTS

Dasty3 relies on a modular architecture, which facilitates development of specialized plug-ins. Dasty3 like Dasty2 is a web-based DAS client. However, Dasty3 is not just a client, but a platform delivering a rich API over which tailored plug-ins can be built. Dasty3 inherits from Dasty2 the modularity of its Graphical User Interface (GUI). Additionally, it provides a Web-based framework for visualizing and manipulating information from DAS sources as well as other third-party data providers. It facilitates delivering unified views upon which several manipulation facilities can be implemented. It offers a public Application Programming Interface (API) that delivers methods for integrating, visualizing and manipulating data sources.



UNIPROTKB_P03973_SIGNAL_

Feature ID : UNIPROTKB_P03973_SIGNAL_1_25
Feature Label : UNIPROTKB_P03973_SIGNAL_1_25
Type : signal_peptide
Type ID : SO:0000418
Category : inferred by curator (ECO:0000001)
Method : UniProt
Start : 1
End : 25
Score : 0.0
Orientation : 0
Phase : -

The annotation is in accordance with the version of the protein sequence.
Caution! The annotation may refer to an old version of the protein sequence, so the position of features may be incorrect.
Group of features classified by the annotation server.
Features grouped in the same line by Dasty3.

Fig. 1. A view of Dasty3 and its plug-ins.

2.1 User experience

As illustrated in Figure 1, Dasty3 preserves the look and feel of Dasty2; in this way, the learning curve is minimized. Dasty3 enhances some of the previous functionalities available in Dasty2 and adds new ones. For instance, the search plug-in allows users to define the sources against which the query should be executed.

Dasty3 uses PICR (Côté *et al.*, 2007) for matching the query string to the corresponding UniProt accession or identifier; the connection to the UniProt DAS reference server (Jones *et al.*, 2005) is then established and the sequence retrieved. Finally, it queries all the selected sources and merges the retrieved sequence annotations. The set of predefined plug-ins provides the user with a unified, organized and interactive view of the available data. A query retrieving information for protein P03973 (antileukoproteinase) is illustrated

[*]To whom correspondence should be addressed.

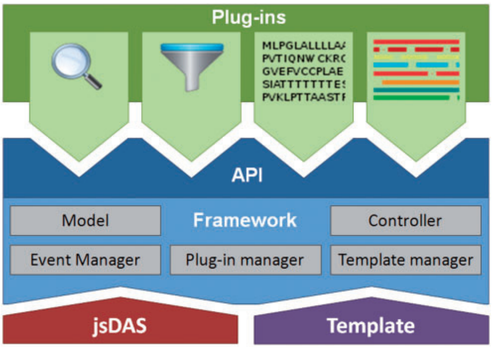


Fig. 2. Dasty3 architecture.

in Figure 1. The 3D view of the protein structure, supported by Jmol (http://jmol.sourceforge.net), is displayed on the right-hand side while the features, organized in tracks, are presented at the bottom. The pop-up window presents a detailed view of the annotation that is highlighted on the sequence.

New data sources can be manually added including the URL of a valid DAS source. DAS sources can also be selected from a list of sources provided by the DAS registry. The filtering plug-in toggles the display of protein annotations. New filters based on DAS-related ontologies, such as those available from the Ontology Look-up service (Côté *et al.*, 2008), have been included to consistently filter annotations from different sources. A new interaction plug-in retrieves a summary for molecular interaction data. We are using the DAS Writeback (Salazar *et al.*, 2011) to support community annotation. The DAS Writeback is an extension to the DAS protocol. It allows adding, editing and deleting annotations. These functionalities have been implemented in Dasty3 over the DAS Writeback plug-in. Dasty3 also makes it possible to save snapshots of user's sessions in the form of a JSON file. For instance, users may search, select and highlight a feature in the sequence; then, they can generate the corresponding snapshot of the session for future analysis or for sharing it with collaborators.

2.2 Architecture

Dasty3 is based on a plug-in architecture. It emphasizes extensibility, plug-in customization and interoperability e.g. data exchange among plug-ins. For instance, information is exchanged between the sequence and the 3D structure plug-in; as amino acids are selected, these are highlighted in the 3D structure. The architecture makes it possible for plug-ins to listen events from other plug-ins. Furthermore, the architecture also makes it easy for developers to integrate Dasty3 into existing Web applications as well as to configure the look-and-feel by defining customized templates.

As illustrated in Figure 2, Dasty3 brings together several components that can be accessed by using the API. jsDAS (http://code.google.com/p/jsdas/) is being used as a bridge between the framework and DAS servers; it automatically generates JavaScript objects representing, in concordance with the DAS 1.6 specification, the DAS responses. The framework has five major components, namely: (i) the Model, defining the DAS objects according to the DAS 1.6 specification (http://www.biodas.org); (ii) the Controller, managing the retrieval, organization and storage of the data; (iii) the Plug-in Manager, making it possible to load plug-ins; (iv) the Event Manager, managing and triggering events so that plug-ins can communicate with each other; and, (v) the Template Manager, rendering the plug-ins in the available area according to the selected template. Plug-ins are components that extend the capabilities of the framework by delivering a particular functionality. They are defined as a tightly coupled collection of JavaScript and CSS files. Templates add flexibility to the Graphical User Interface (GUI) by simplifying the display of plug-ins and customization of the layout. For instance, moving and reorganizing plug-ins according to end user preferences can be defined in a template. The modularization upon which Dasty3 relies facilitates sharing; in the same way, as developers share modules and themes in Content Management Systems (CMSs) such as Drupal (http://drupal.org) and Joomla (http://www.joomla.org), they are able to share templates and plug-ins in Dasty3.

3 FINAL REMARKS

Dasty3 is the first DAS client fully compliant with the 1.6. DAS specification. It has been tested on Internet Explorer, Firefox, Chrome and Safari. Dasty3 is highly modular and extensible, and new components can be easily added to the framework. The interoperability delivered by the API facilitates the flow of data across plug-ins. The architecture also facilitates the organization of the front-end by means of templates; easing in this way the definition of the layout and improving the user experience. Dasty3 separates the functional components, being managed by the Plug-in manager, from those related to the user experience and graphical layout, being managed by the Template manager. This separation makes it easier for developers to deliver richer applications built over the Dasty3 framework; in the same way, end users receive more personalized task-oriented tools. In addition, Dasty3 facilitates mechanisms for integrating resources beyond the DAS ecosystem; by doing so, the retrieved information presents a broader view to the end user.

Funding: NHLBI Proteomics Center Award HHSN268201000035C, as well as the European Commission grants ENFIN (LSHG-CT-2005-518254); APO-SYS (FP7-HEALTH-2007-200767) in part.

Conflict of Interest: none declared.

REFERENCES

Côté,R. *et al.* (2008) The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. *Nucleic Acids Res.*, **36**, W372–W376.

Côté,R. *et al.* (2007) The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. *BMC Bioinformatics*, **8**, 401.

Jenkinson,A. *et al.* (2008) Integrating biological data - the distributed annotation system. *BMC Bioinformatics*, **9**, S3.

Jimenez,R.C. *et al.* (2008) Dasty2, an Ajax protein DAS client. *Bioinformatics*, **24**, 2119–2121.

Jones,P. *et al.* (2005) Dasty and UniProt DAS: a perfect pair for protein feature visualization. *Bioinformatics*, **21**, 3198–3199.

Prlic,A. *et al.* (2005) Adding some SPICE to DAS. *Bioinformatics*, **1**, ii40–ii41.

Salazar,G. *et al.* (2011) DAS Writeback: a collaborative annotation system. *BMC Bioinformatics*, **12**, 143.

Curriculum Vitae

Bernat Gel Moreno

☎ *627 92 99 15*
✉ *bernatgel@gmail.com*

Education

2006–present **PhD. Software**, *Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya*, Barcelona.

2005–2006 **Master in Computing**, *Universitat Politècnica de Catalunya*, Barcelona.

2000–2005 **Enginyeria Informàtica**, *Universitat Politècnica de Catalunya*, Barcelona.

Experience

2011–present **Bioinformatician**, *Hereditary Cancer Program. Institut de Medicina Predictiva i Personalitzada del Càncer. IMPPC.*, Badalona, Barcelona.

2005–present **PhD Student in Bioinformatics**, *Departament de Llenguatges i Sistemes Informàtics.*, Barcelona.

2004–2010 **Bioinformatician and Biostatician**, *Laboratori d'Oncologia Molecular del Departament d'Anatomia i Embriologia de la Facultat de Medicina-Clínic de la Universitat de Barcelona*, Barcelona.

International

2009–2010 **Research Stay**, *European Bioinformatics Institute (EBI)*, Hinxton, Cambridge, UK.
As part of my PhD I stayed for half a year working at the EBI in further developments of DAS.

2004–2004 **Erasmus**, *Aberdeen University*, Aberdeen, Scotland.
Taking part on the Erasmus promgramme, I spent 6 months in Aberdeen, UK, taking regular CS courses.

Projects

PhD. Thesis

Title *Dissemination and Visualisation of Biological Data*

Supervisor Dr. Xavier Messeguer Peypoch

Description Development of new tools (mainly web applications) to share and integrate biological data via DAS (Distributed Annotation System). The developed tools include an interactive genome browser (http://gralggen.lsi.upc.edu/recerca/genexp/) and an automatic creator of DAS sources (http://www.ebi.ac.uk/panda-srv/easydas/).

Master Thesis

Title *Genome Map: a web application for annotated genome visualization*

Supervisor Dr. Xavier Messeguer Peypoch

Description Development of the first prototype of an interactive web-based genomic viewer using the newest (at that moment) web technologies.

Degree Final Project

Title *Cerca de gens diana dels micorRNAs del virus d'Epstein-Barr* (*Seaching target human genes for the microRNAs encoded by the Epstein-Barr virus*)

Supervisors Dr. Xavier Messeguer Peypoch and Dr. Marià Monzó Planella

Description We developed a system to predict the human targets of the EBV microRNAs based on sequence similarity and clustering of targets in pathways.

Grants and Awards

FI FI predoctoral fellow from 1-01-2007 al 31-12-2010.

BE BE travel grant for a six months stage at the European Bioinformatics Institute.

IMIM I received a grant from Institut Municipal d'Investigacions Mèdiques (IMIM) to work on the FIS "Estudio farmacogenómico de los polimorfismos en los genes reparadores del DNA en pacientes afectados de un cáncer de cabeza y cuello".

BDigital Winner, with my PhD advisor Xavier Messeguer, of the "Premi BDdigital Global Congress Ciutat del Coneixement" with our project GenomPort, based on the GenExp genome browser.

Teaching

2008-2010 **Information Recovery**, *Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya*, Barcelona.
For three quatrimesters I've been teaching Information Recovery (Recuperació de la Informació, RI) at Facultat d'Informàtica de Barcelona.

Publications

[1] Alfons Navarro, Carmen Muñoz, Anna Gaya, Marina Díaz-Beyá, Bernat Gel, Rut Tejero, Tania Díaz, Antonio Martinez, and Mariano Monzó. MiR-SNPs as Markers of Toxicity and Clinical Outcome in Hodgkin Lymphoma Patients. *PLoS ONE*, 8(5):e64716, May 2013.

[2] Eric P Rahrmann, Adrienne L Watson, Vincent W Keng, Kwangmin Choi, Branden S Moriarity, Dominic a Beckmann, Natalie K Wolf, Aaron Sarver, Margaret H Collins, Christopher L Moertel, Margaret R Wallace, Bernat Gel, Eduard Serra, Nancy Ratner, and David a Largaespada. Forward genetic screen for malignant peripheral nerve sheath tumor formation identifies new genes and pathways driving tumorigenesis. *Nature genetics*, (May):1–13, May 2013.

[3] Marina Díaz-Beyá, Alfons Navarro, Gerardo Ferrer, Tania Díaz, Bernat Gel, M Camós, M Pratcorona, M Torrebadell, Maria Rozman, Dolors Colomer, Mariano Monzo, and J Esteve. Acute myeloid leukemia with translocation (8;16)(p11;p13) and MYST3-CREBBP rearrangement harbors a distinctive microRNA signature targeting RET proto-oncogene. *Leukemia : official journal of the Leukemia Society of America, Leukemia Research Fund, U.K*, October 2012.

[4] Carles Garcia-Linares, Jaume Mercadé, Bernat Gel, Josep Biayna, Ernest Terribas, Conxi Lázaro, and Eduard Serra. Applying microsatellite multiplex PCR analysis (MMPA) for determining allele

copy-number status and percentage of normal cells within tumors. *PloS one*, 7(8):e42682, January 2012.

[5] Jose M. Villaveces, Rafael C. Jimenez, Leyla J Garcia, Gustavo a. Salazar, Bernat Gel, Nicola Mulder, Maria Martin, Alexander Garcia, and Henning Hermjakob. Dasty3, a WEB framework for DAS. *Bioinformatics (Oxford, England)*, 27(18):2616–7, September 2011.

[6] Bernat Gel Moreno and Xavier Messeguer Peypoch. GenExp: An Interactive Web-Based Genomic DAS Client with Client-Side Data Rendering. *PLoS ONE*, 6(7):e21270, July 2011.

[7] Marc Campayo, Nuria Viñolas, Alfons Navarro, Enric Carcereny, Francesc Casas, Bernat Gel, Tania Diaz, Josep Maria Gimferrer, Ramon M Marrades, Jose Ramirez, and Mariano Monzo. Single nucleotide polymorphisms in tobacco metabolism and DNA repair genes and prognosis in resected non-small-cell lung cancer. *The Journal of surgical research*, 167(1):e5–12, May 2011.

[8] Tania Diaz, Alfons Navarro, Gerardo Ferrer, Bernat Gel, Anna Gaya, Rosa Artells, Beatriz Bellosillo, Mar Garcia-Garcia, Sergi Serrano, Antonio Martínez, and Mariano Monzo. Lestaurtinib inhibition of the Jak/STAT signaling pathway in hodgkin lymphoma inhibits proliferation and induces apoptosis. *PloS one*, 6(4):e18856, January 2011.

[9] Bernat Gel Moreno, Andrew M Jenkinson, Rafael C Jimenez, Xavier Messeguer Peypoch, and Henning Hermjakob. easyDAS: Automatic creation of DAS servers. *BMC bioinformatics*, 12(1):23, January 2011.

[10] Rafael C. Jimenez, Gustavo a. Salazar, Bernat Gel, Joaquin Dopazo, Nicola Mulder, and Manuel Corpas. myKaryoView: a light-weight client for visualization of genomic data. *PloS one*, 6(10):e26345, January 2011.

[11] Alfons Navarro, Tania Diaz, Elena Gallardo, Nuria Viñolas, Ramon M Marrades, Bernat Gel, Marc Campayo, Angels Quera, Eva Bandres, Jesus Garcia-Foncillas, Jose Ramirez, and Mariano Monzo. Prognostic implications of miR-16 expression levels in resected non-small-cell lung cancer. *Journal of surgical oncology*, (December 2010):411–415, December 2010.

[12] Rosa Artells, Isabel Moreno, Tania Díaz, F Martínez, Bernat Gel, Alfons Navarro, Rafael Ibeas, J Moreno, and Mariano Monzó. Tumour CD133 mRNA expression and clinical outcome in surgically resected colorectal cancer patients. *European journal of cancer (Oxford, England : 1990)*, 46(3):642–9, February 2010.

[13] Elena Gallardo, Alfons Navarro, Nuria Viñolas, Ramon M Marrades, Tania Diaz, Bernat Gel, Angels Quera, Eva Bandres, Jesus Garcia-Foncillas, Jose Ramirez, and Mariano Monzo. miR-34a as a prognostic marker of relapse in surgically resected non-small-cell lung cancer. *Carcinogenesis*, 30(11):1903–1909, November 2009.

[14] Aina Pons, Benet Nomdedeu, Alfons Navarro, Anna Gaya, Bernat Gel, Tania Diaz, Sandra Valera, María Rozman, Mohamed Belkaid, Emili Montserrat, and Mariano Monzo. Hematopoiesis-related microRNA expression in myelodysplastic syndromes. *Leukemia & Lymphoma*, 50(11):1854–1859, November 2009.

[15] Alfons Navarro, Tania Diaz, Antonio Martinez, Anna Gaya, Aina Pons, Bernat Gel, Carles Codony, Gerardo Ferrer, Carmen Martinez, Emili Montserrat, and Mariano Monzo. Regulation of JAK2 by miR-135a: prognostic impact in classic Hodgkin lymphoma. *Blood*, 114(14):2945–2951, October 2009.

[16] Bernat Gel and Xavier Messeguer. Implementing an Interactive Web-Based DAS Client. In Juan Corchado, Juan De Paz, Miguel Rocha, and Florentino Fernández Riverola, editors, *2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008)*, volume 49 of *Advances in Soft Computing*, pages 83–91. Springer Berlin / Heidelberg, 2009.

[17] Miquel Granell, Alvaro Urbano-Ispizua, Aina Pons, Juan Ignacio Aróstegui, Bernat Gel, Alfons Navarro, Sonia Jansa, Rosa Artells, Anna Gaya, Carme Talarn, Francesc Fernández-Avilés, Carmen Martínez, Montserrat Rovira, Enric Carreras, Ciril Rozman, Manel Juan, Jordi Yagüe, Emili Montserrat, and Mariano Monzó. Common variants in NLRP2 and NLRP3 genes are strong

prognostic factors for the outcome of HLA-identical sibling allogeneic stem cell transplantation. *Blood*, 112(10):4337–4342, November 2008.

[18] Mariano Monzo, Alfons Navarro, Eva Bandres, Rosa Artells, Isabel Moreno, Bernat Gel, Rafael Ibeas, Jose Moreno, Francisco Martinez, Tania Diaz, Antonio Martinez, Olga Balagué, and Jesus Garcia-Foncillas. Overlapping expression of microRNAs in human embryonic colon and colorectal cancer. *Cell Research*, 18(8):823–833, August 2008.

[19] Alfons Navarro, Anna Gaya, Antonio Martinez, Alvaro Urbano-Ispizua, Aina Pons, Olga Balagué, Bernat Gel, Pau Abrisqueta, Armando Lopez-Guillermo, Rosa Artells, Emili Montserrat, and Mariano Monzo. MicroRNA expression profiling in classic Hodgkin lymphoma. *Blood*, 111(5):2825–2832, March 2008.

[20] Isabel Moreno-Solórzano, Rafael Ibeas-Rollan, Mariano Monzó-Planella, José Moreno-Solórzano, Francisco Martínez-Ródenas, Edmon Pou-Sanchis, Raquel Hernández-Borlan, Marta Navarro-Vigo, Silvia Ortigosa-Rodríguez, and Bernat Gel-Moreno. Two Doses of oxaliplatin with capecitabine (XELOX) in metastatic colorectal cancer. *Clinical colorectal cancer*, 6(9):634–40, September 2007.

[21] Mariano Monzo, Isabel Moreno, Alfons Navarro, Rafael Ibeas, Rosa Artells, Bernat Gel, Francisco Martinez, Jose Moreno, Raquel Hernandez, and Marta Navarro-Vigo. Single nucleotide polymorphisms in nucleotide excision repair genes XPA, XPD, XPG and ERCC1 in advanced colorectal cancer patients treated with first-line oxaliplatin/fluoropyrimidine. *Oncology*, 72(5-6):364–370, 2007.

[22] Joan Carles, Mariano Monzo, Marta Amat, Sonia Jansa, Rosa Artells, Alfons Navarro, Palmira Foro, Francesc Alameda, Angel Gayete, Bernat Gel, Maribel Miguel, Joan Albanell, and Xavier Fabregat. Single-nucleotide polymorphisms in base excision repair, nucleotide excision repair, and double strand break genes as markers for response to radiotherapy in patients with Stage I to II head-and-neck cancer. *International journal of radiation oncology, biology, physics*, 66(4):1022–30, November 2006.