BMC Bioinformatics

SOFTWARE Open Access

ISAAC - InterSpecies Analysing Application using Containers

Herbert Baier and Jörg Schultz*

Abstract

Background: Information about genes, transcripts and proteins is spread over a wide variety of databases. Different tools have been developed using these databases to identify biological signals in gene lists from large scale analysis. Mostly, they search for enrichments of specific features. But, these tools do not allow an explorative walk through different views and to change the gene lists according to newly upcoming stories.

Results: To fill this niche, we have developed ISAAC, the InterSpecies Analysing Application using Containers. The central idea of this web based tool is to enable the analysis of sets of genes, transcripts and proteins under different biological viewpoints and to interactively modify these sets at any point of the analysis. Detailed history and snapshot information allows tracing each action. Furthermore, one can easily switch back to previous states and perform new analyses. Currently, sets can be viewed in the context of genomes, protein functions, protein interactions, pathways, regulation, diseases and drugs. Additionally, users can switch between species with an automatic, orthology based translation of existing gene sets. As todays research usually is performed in larger teams and consortia, ISAAC provides group based functionalities. Here, sets as well as results of analyses can be exchanged between members of groups.

Conclusions: ISAAC fills the gap between primary databases and tools for the analysis of large gene lists. With its highly modular, JavaEE based design, the implementation of new modules is straight forward. Furthermore, ISAAC comes with an extensive web-based administration interface including tools for the integration of third party data. Thus, a local installation is easily feasible. In summary, ISAAC is tailor made for highly explorative interactive analyses of gene, transcript and protein sets in a collaborative environment.

Keywords: Teamwork, Gene sets, Explorative analyses, Cross-species analyses

Background

Over the last 10 to 15 years, biology has changed into a ‘more precise and quantitative science’ [1]. New high throughput technologies generate data covering different aspects of molecules in an ever increasing pace. As a result, we are now drowning in data when looking for biological stories. Accordingly, bioinformatics methods and databases to deal with this flood of information have been developed. Whereas in the beginning these computational tools were available mainly to bioinformaticians, many tools and databases are nowadays accessible via the web and can be interrogated also by non-computational trained biologists. But, there are still some challenges to cope with when trying to find biological meaning within this flood of data. First, different types of data are distributed over a wide variety of databases and web-based resources. For example a biologist will have to go to Ensembl [2] or the UCSC genome browser [3] when searching for genomic information. Next she might look up functional information in the GeneOntology [4] (which generously has been integrated in a multitude of other tools and databases). If especially interested in disease genes, OMIM [5] and DrugBank [6] might be useful resources. Next, to identify functionally related genes, databases like KEGG [7], STRING [8], or in more specific cases mirRBase [9] might be questioned. The challenge of distributed data has been addressed by different higher level tools. These focus mainly on the evaluation of larger datasets generated by high throughput methods and the more or less automated annotation and statistical evaluation of these gene

* Correspondence: joerg.schultz@biozentrum.uni-wuerzburg.de
Department of Bioinformatics, Biocenter, University of Würzburg, Am Hubland, Würzburg 97074, Germany

sets. An outstanding example is DAVID [10,11]. It provides a variety of functional annotation tools (like gene enrichment analysis, pathway mapping), gene accession conversion, a genome browser and a stateful web service [12]. Gene lists can be uploaded in different identifier formats and sub lists can be created during the enrichment analysis. Furthermore, the lists can be renamed, removed, combined and downloaded. Related functional profiling tools are GEPAT [13], Onto-Express [14], Onto-Tools [15], FACT [16] BABELOMICS [17], FatiGO + [18], GeneTrail [19], g:Profiler [20], VisANT [21], Reactome [22], MAPPFinder [23], GFINDer [24], GOLEM [25]. Provides a good overview on enrichment tools [26]. The Ingenuity System [27] is a commercial software that is widely used to analyze and model complex biological and chemical systems. Finally, Cytoscape [28] is a generic tool for network analysis and visualization whose network information can be associated with gene expression data.

As mentioned above, the main goal of these tools is the statistical evaluation and functional characterization of given, mostly large, gene sets. Thus it is in the nature of these tools, that the user is not allowed to interactively change the gene lists within one analysis. For their application, this makes perfect sense, as these tools provide a reproducible annotation pipeline. But, there is a different type of user who might be more interested in the explorative analysis of smaller gene sets. She might start with a few genes, analyze them under one aspect and find other genes of interest. Now she might want to extend the gene sets and analyze the new list under a different aspect. WebGestalt [29,30] did a first step into this direction. Here, different sets could be merged, but the manual addition of genes is not possible. However, in the current online version of WebGestalt these set operations are missing. Complementary, WhichGenes [31] enables generating gene sets based on various sources and to combine these sets. Thus, sets of genes involved in glycolysis and encoded on a specific chromosome can be generated. Still, it does not allow viewing one gene or gene set under different biological aspects or performing analyses on sets. Thus, we wanted to create a tool which integrates the main idea of enrichment tools, namely to analyze gene lists under a wide variety of functional aspects, with the ability to manually add and delete sets of interesting genes to enable explorative analyses. As the amount and detail of functional information differs widely between different species, we also wanted to enable cross species analysis. We have implemented these ideas in the Web based tool ISAAC (http://isaac.bioapps.biozentrum.uni-wuerzburg.de), an acronym for 'InterSpecies Analysing Application using Containers'.

Implementation

Object oriented strategy

Traditionally, the selection of gene sets follows a procedural approach. An example might be the BioMart interface to the Ensembl databases [32]. In the most straightforward scenario, the user first selects a species/dataset, then defines filters for the gene sets and chooses which attributes should be reported. Finally, the corresponding set is calculated. If the user wants to add other filters, the procedure has to be changed and a new set is calculated. Contrasting ISAAC uses a more object oriented strategy. Its central point of view is the information that the biologist wants to analyze and his/her knowledge. The information is represented as sets which can contain genes, transcripts and proteins (the objects) which then can be compared and/or modified (the methods) (Figure 1). All elements in a set belong to the same species and the following transitive property is satisfied: if a protein belongs to a set, then the coding transcript belongs to the set, and if a transcript belongs to a set, then the coding gene belongs to the set. Formally, let p be a protein, t be a transcript and S be a set of genes, transcripts and proteins of a species, then

$$\begin{aligned} p \in S &\rightarrow t_p \in S \\ t \in S &\rightarrow g_t \in S \end{aligned}$$

where t_p is the transcript of the protein p and g_t is the gene of the transcript t. The reverse property is not required, i.e. a gene can be part of the set without adding its transcripts or proteins. It is assured that the sets are consistent at all times. Biologically, this transitivity is of importance as it enables the integration of information focusing on different biological entities. A user can add a specific splice variant of a gene, i.e. a protein, to a set. Automatically, the corresponding transcript as well as the coding gene is added to the set. Therefore, information related to the gene like a disease association can be analyzed. Still, when going back to the protein level only the specific isoform is considered. Complementary, if the user adds a gene to a set no transcript or protein information is added by default as further information might be specific to one or a few of the isoforms of the gene. If desired, all transcripts and proteins encoded by a gene can be added to the set, thus ensuring the consistency of the sets. The basic set comparisons (equal =, proper subset ⊂, subset ⊆) and operations (union ∪, intersection ∩, set minus \) are supported. Hence, elements can be added using the union operator and removed using the set minus operator. Due to the transitivity, the set comparisons and the set minus operation are performed on a selected level, namely genes, transcripts or proteins. Additionally, sets can be created, copied, cleared, removed, imported and exported.

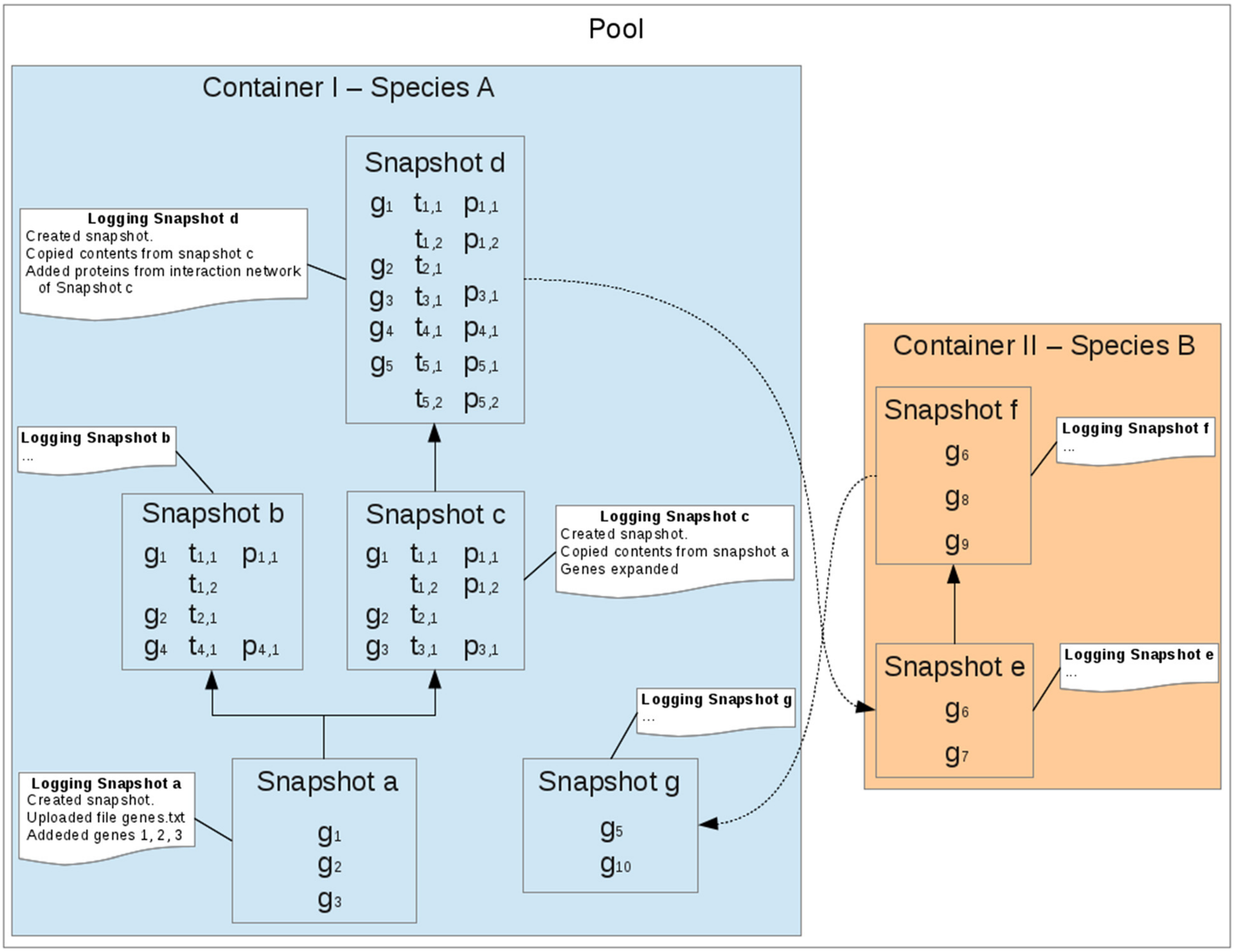


Figure 1 ISAAC core concepts – Let $t_{a,b}$ be the transcript coding for the protein $p_{a,b}$ and g_a be the gene coding for the transcript $t_{a,b}$. A user creates the snapshot a in the container I for species A and uploads a file containing three genes, namely g_1, g_2 and g_3. From snapshot a, the user creates two child snapshots b and c (the contents of the snapshot a are replicated to the snapshots b and c). In the snapshot b the user adds the proteins $p_{1,1}$ and $p_{4,1}$ (due to the transitive rule, $t_{1,1}$, $t_{4,1}$ and g_4 are also added) and the transcripts $t_{1,2}$ and $t_{2,1}$ and removes the gene g_3. In the snapshot c the genes are extended, this means their transcripts and proteins are added. From snapshot c the child snapshot d is created. All proteins of the interaction network containing all proteins which directly interact with all proteins in the snapshot c, namely $p_{1,1}$, $p_{1,2}$, $p_{3,1}$, $p_{4,1}$, $p_{5,1}$ and $p_{5,2}$ are added. Again, corresponding transcripts and genes are added automatically. Next, the snapshot e in the container II for species B is created and the orthologous genes of snapshot d are imported. From snapshot e the user creates the child snapshot f and adds the genes g_8 and g_9 and removes the gene g_7. Finally, the snapshot g is created in the container I and the orthologous genes of snapshot f are included.

A version control system manages the sets in a tree structure that allows biologists to keep track of different versions. Furthermore, for each action a history is logged enabling the tracing of changes of sets. In ISAAC context a set configuration managed by the version control system is called snapshot and a container is a collection of snapshots. Each snapshot belongs to exactly one container and all snapshots in a version tree belong to the same container. At any time, a user can go back to an older snapshot and use it as the starting point for a new analysis by generating a new child snapshot. Thus, a tree like structure of analyses can be generated. Special properties can be defined in a container, such as a description, color and comments.

This core system can now be used from different modules, which mainly perform analyses on sets, modify sets with the given methods and visualize sets and results of analyses. Thus, complex biological analyses covering different biological aspects are broken down into independent, interchangeable modules. The resulting non-linear application flow supports the biologist in searching for biological stories in their data.

Java EE technology

ISAAC is implemented in Java EE 6 (Java Platform, Enterprise Edition) and uses the web component JSF 2.0 (Java Server Faces) to generate dynamic web pages with Ajax support. This technology substantially simplifies the development of an application, since it creates standardized, reusable modular components and enables the tier to handle many aspects of programming automatically like persistence, messaging and security. Hence, a Java application server is required to run ISAAC. ISAAC was developed and tested using JBoss application server. The strict client/server architecture allows multiple frontend clients to be developed and integrated in a standard and easy way, since the process logics are performed on the server. In ISAAC, there is no time out for a client web session. As long the web page is open, its session is held on the web server.

As aforementioned each module contains everything necessary to perform the desired functionality and, therefore, information has to be imported from other sources. However, this information is not tied to specific sources. Each module provides well-defined interfaces and any source fulfilling their requirements can be used. The development of new modules and even web services is therefore straightforward.

Module processes requiring large computer resources are started in background, which avoids blocking the clients till the processes are finished. As soon as a process is finished, the owner is notified within the web interface or, if desired, also via E-Mail. The processes' results are held in the private pools and can be recovered as needed.

Team work capabilities

Today, many research groups are embedded in larger teams. Frequently, different groups work on related aspects of a biological phenomenon using different model species. Therefore, ISSAC as a multi-user system supports teamwork. Each user owns private pools of (i) containers with sets of genes, transcripts and proteins and (ii) results of analysis. Furthermore, pools shared within a group of users can be created. Access to the pools comes in two authorization levels: (i) in the normal level a user is only allowed to read the pool data and (i) in the coordinator level a user is also allowed to update pools and to add/remove users from the group. Special groups can be defined, which allow all users to access their pools in a normal level. In addition to sharing containers, coordinators can also place processes in group pools enabling access to all group members.

Results and discussion

Available modules

An overview of currently implemented modules is given in Table 1. Obviously, the core of a system to analyze lists of genes has to hold information about genes, transcripts and proteins. This is handled in the **genome module**. On start the chromosomes of the selected species are displayed. Here, regions can be selected and genes within this region are shown. These can either directly be added to a snapshot or further inspected in the 'gene view'. Here, genes, transcripts, proteins, associated diseases, drugs and miRNAs are shown, among other information. Transcripts and proteins can be selected and added to snapshots. For proteins, Interpro annotations are provided, allowing a first functional characterization [33]. Snapshots can be selected and their contents will be highlighted in the chromosome view using the respective container color. As ISAAC also provides information about orthologous relations, the user can switch between different species automatically 'translating' the gene sets. Further genomes and enzyme activities can be imported from Ensembl [2,34] and UniProt [35], respectively, via the administration interface.

In a cell, no protein works on its own. Thus, to understand the function of a single or sets of proteins, one always has to consider their interaction partners. To allow analyzing sets in this context, we implemented the **protein interaction module**. Starting with direct interactions, it can be extended to show higher level interactions. Although data from any protein interaction database can be integrated, we currently imported data from STRING [8]. Thus, interactions are annotated on the gene level, although in the cell proteins are interacting. As ISAAC ensures consistent sets with gene and

Table 1 Currently implemented modules

Module	Datasource(s)	Usage
Genome	Ensembl [2], UniProt [35]	Search for genes, transcripts and proteins and add them to sets. Features are visualized.
Protein Interaction	STRING [8]	Analyze protein interactions within a set. Identify interacting proteins and add them to sets.
Pathway	KEGG [7]	Analyze genes in sets in a metabolic and pathways context. Add further genes of a pathway to sets.
GO enrichment	GeneOntology [4]	Functional characterization of sets. Extend sets based on function.
microRNA	TarBase [38]	Reveal microRNA based regulation of genes in sets. Search for genes regulated by specific microRNAs.
Disease	OMIM [5]	Identify mendelian disease genes in sets. Search for genes associated with a mendelian disease.
Drug	DrugBank [6]	Identify drug targets in sets. Search for genes affected by a drug.
Orthology	Ensembl [2]	Orthology based translation of sets to other species.
Team Work	-	Share sets between users and within groups.

transcript automatically added, this distinction is hidden from the user. In the graph, single genes of interest or all genes can be added to a snapshot. Again, STRING data can be imported via the administration interface.

The protein interaction module enables viewing sets in the context of networks. Still, the type of interaction is not detailed out. The biological **pathway module** allows the graphical visualization of enzymes belonging to a snapshots of a species in their pathway context [7]. In a pathway diagram the EC-numbers are highlighted in three different ways: (i) the enzymatic function is covered by the selected snapshots, (ii) the species contains an enzyme with the EC classification but it is not part of the snapshots and (iii) no protein with the EC number is annotated in the species. As usual, genes coding for a given enzymatic function or the whole pathway can be added to snapshots. Furthermore, an enzyme can be selected going to the protein's view of the genome module.

To enable a fast functional characterization of genes, transcripts and proteins in sets, we implemented the **GO term enrichment module**. Based on GeneOntology [4], the biological processes, cellular components and molecular functions of proteins/genes of snapshots can be analyzed and displayed. To improve the tree based presentation of the directed acyclic GO graph, nodes with more than one parent are replicated and only sub trees with at least one match are shown. For each node the following information is given: (i) the GO description, (ii) the number of proteins in the selected snapshot(s) belonging to this node, (iii) the total number of proteins in the genome belonging to this node, (iv) the number of proteins of the selected snapshot(s) belonging to the sub tree rooted in this node, (v) the total number of proteins in the selected snapshot and (vi) the p-value of this node (parent–child-union approach of the hypergeometric distribution [36,37]). Proteins belonging to a node or its sub tree can be added to snapshots. On the other direction, for each selected protein a list of GO identifiers is given. A protein can be selected going to the protein's view of the genome module. Furthermore, a GO identifier can be selected which highlights the paths to the root (a node can coexist more than once in the tree). All GO data can be imported via the web based administration interface.

To get insights about possible regulatory mechanisms of genes in a snapshot, the **microRNA module** was implemented. It supports the search for microRNAs and lists genes regulated by the specified microRNA, which can be added to a snapshot. Complementary, all microRNAs regulating genes in selected snapshots can be listed. Information about microRNA was imported from TarBase [38].

Finally, we enable to search for genes associated with a disease and genes which are known drug targets in the **disease module** and the **drug module**, respectively. Again, a user can start with a disease or a drug, get information about involved genes and add them to the snapshot. Alternatively she can list all diseases associated with genes in snapshots and drugs affecting these genes. The disease module supports external links to the OMIM database [5] and the drug module to DrugBank [6]. Tools are provided to import OMIM information from the Ensembl (BioMart) database and drugs from DrugBank.

One of the main ideas behind the development of ISAAC was to carry out analyses across species boundaries. This is enabled by adding **orthology** information. Here, the user can switch between different species and the actual snapshot is 'translated' to the new species. For administration, an interface was implemented to import orthologous data from TSV files created by e.g. BioMart [34].

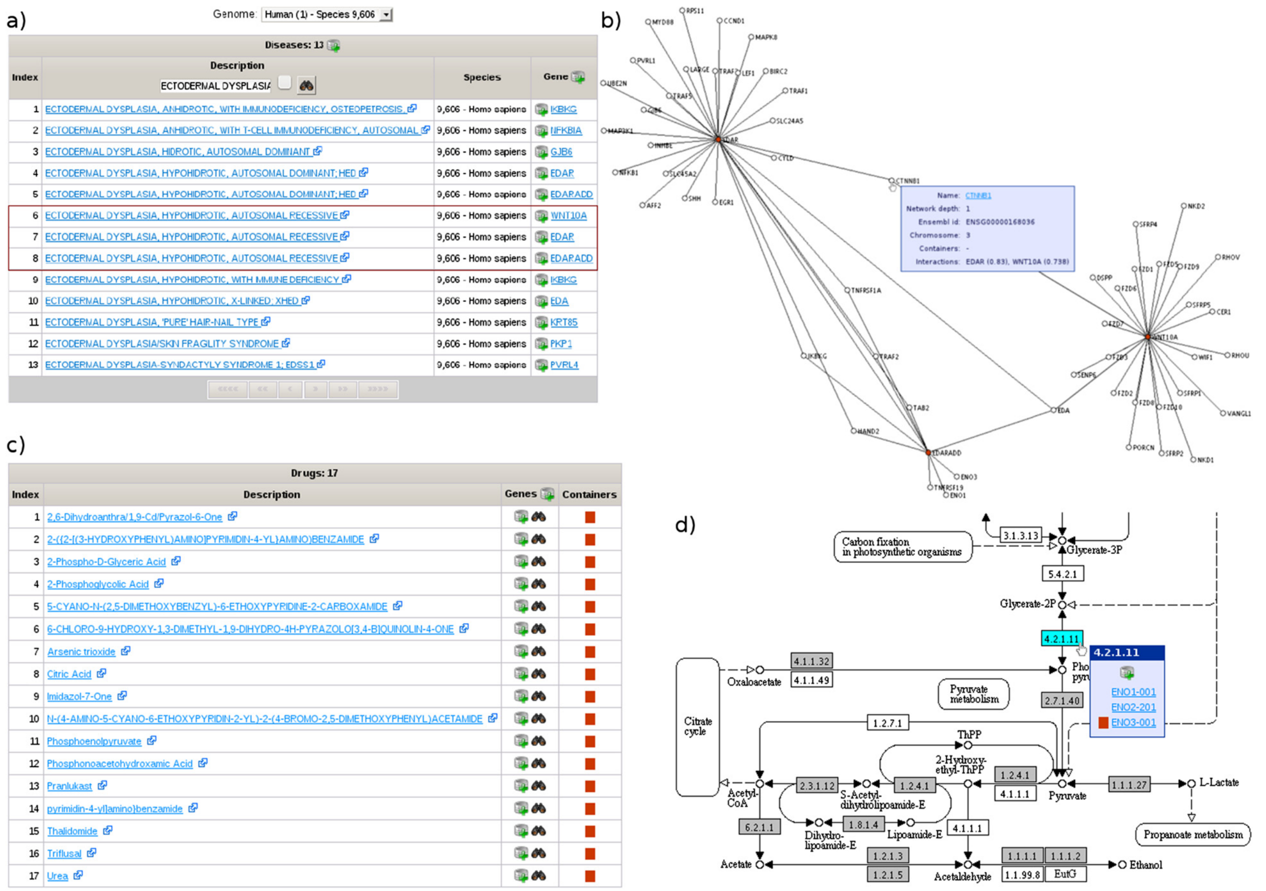
Use cases

Different aspects of genes/proteins

As ISAAC includes information from OMIM, a user can look up a disease, for example ectodermal dysplasia. OMIM Entries and the associated genes are displayed (Figure 2a). Focusing on the recessive autosomal variant, she creates a snapshot with the three associated genes. In a first attempt to search for drug candidates, she can switch to the drug tab and list all known drugs which interact with proteins encoded by genes in the actual snapshot. Here, she will find none. In an attempt to search for further candidate genes, she switches to the protein interaction module. Here, she creates an interaction network containing all genes which directly interact with all genes in the current snapshot, i.e. the disease genes (Figure 2b). In the example case, this network comprises 58 genes, which are added to the snapshot. Now, with this expanded gene set, one can go back to the drug tab and check for drug-able genes. Indeed, one finds that 7 of the directly interacting genes are targets of known drugs (Figure 2c). These now identified candidate genes can be added to a new snapshot. To predict side effects, one can go to the pathway tab and check, in which pathways the genes are involved. Here, one finds one gene, which is involved in glycolysis (Figure 2d). After creating a new snapshot with this gene, one can switch species and identify the mouse ortholog as a candidate mouse model. Obviously, this scenario is rather naïve considering the identification of drug candidate genes, but it should exemplify the possibilities to view gene lists under different biological aspects and how a user can interactively adapt the gene/protein sets.

Non-Model organisms

The increasing pace of genome sequencing results in genomes of experimentally poorly characterized species.



a)

Genome: Human (1) - Species 9,606

Diseases: 13

Index	Description ECTODERMAL DYSPLASIA	Species	Gene
1	ECTODERMAL DYSPLASIA, ANHIDROTIC, WITH IMMUNODEFICIENCY, OSTEOPETROSIS,	9,606 - Homo sapiens	IKBKG
2	ECTODERMAL DYSPLASIA, ANHIDROTIC, WITH T-CELL IMMUNODEFICIENCY, AUTOSOMAL	9,606 - Homo sapiens	NFKBIA
3	ECTODERMAL DYSPLASIA, HIDROTIC, AUTOSOMAL DOMINANT	9,606 - Homo sapiens	GJB6
4	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, AUTOSOMAL DOMINANT; HED	9,606 - Homo sapiens	EDAR
5	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, AUTOSOMAL DOMINANT; HED	9,606 - Homo sapiens	EDARADD
6	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, AUTOSOMAL RECESSIVE	9,606 - Homo sapiens	WNT10A
7	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, AUTOSOMAL RECESSIVE	9,606 - Homo sapiens	EDAR
8	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, AUTOSOMAL RECESSIVE	9,606 - Homo sapiens	EDARADD
9	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, WITH IMMUNE DEFICIENCY	9,606 - Homo sapiens	IKBKG
10	ECTODERMAL DYSPLASIA, HYPOHIDROTIC, X-LINKED; XHED	9,606 - Homo sapiens	EDA
11	ECTODERMAL DYSPLASIA, 'PURE' HAIR-NAIL TYPE	9,606 - Homo sapiens	KRT85
12	ECTODERMAL DYSPLASIA/SKIN FRAGILITY SYNDROME	9,606 - Homo sapiens	PKP1
13	ECTODERMAL DYSPLASIA-SYNDACTYLY SYNDROME 1; EDSS1	9,606 - Homo sapiens	PVRL4

b)

Name: CTNNB1
Network depth: 1
Ensembl id: ENSG00000168036
Chromosome: 3
Containers: -
Interactions: EDAR (0.83), WNT10A (0.738)

c)

Drugs: 17

Index	Description	Genes	Containers
1	2,6-Dihydroanthra/1,9-Cd/Pyrazol-6-One		🟥
2	2-({2-[(3-HYDROXYPHENYL)AMINO]PYRIMIDIN-4-YL}AMINO)BENZAMIDE		🟥
3	2-Phospho-D-Glyceric Acid		🟥
4	2-Phosphoglycolic Acid		🟥
5	5-CYANO-N-(2,5-DIMETHOXYBENZYL)-6-ETHOXYPYRIDINE-2-CARBOXAMIDE		🟥
6	6-CHLORO-9-HYDROXY-1,3-DIMETHYL-1,9-DIHYDRO-4H-PYRAZOLO[3,4-B]QUINOLIN-4-ONE		🟥
7	Arsenic trioxide		🟥
8	Citric Acid		🟥
9	Imidazol-7-One		🟥
10	N-(4-AMINO-5-CYANO-6-ETHOXYPYRIDIN-2-YL)-2-(4-BROMO-2,5-DIMETHOXYPHENYL)ACETAMIDE		🟥
11	Phosphoenolpyruvate		🟥
12	Phosphonoacetohydroxamic Acid		🟥
13	Pranlukast		🟥
14	pyrimidin-4-yl]amino}benzamide		🟥
15	Thalidomide		🟥
16	Triflusal		🟥
17	Urea		🟥

d)

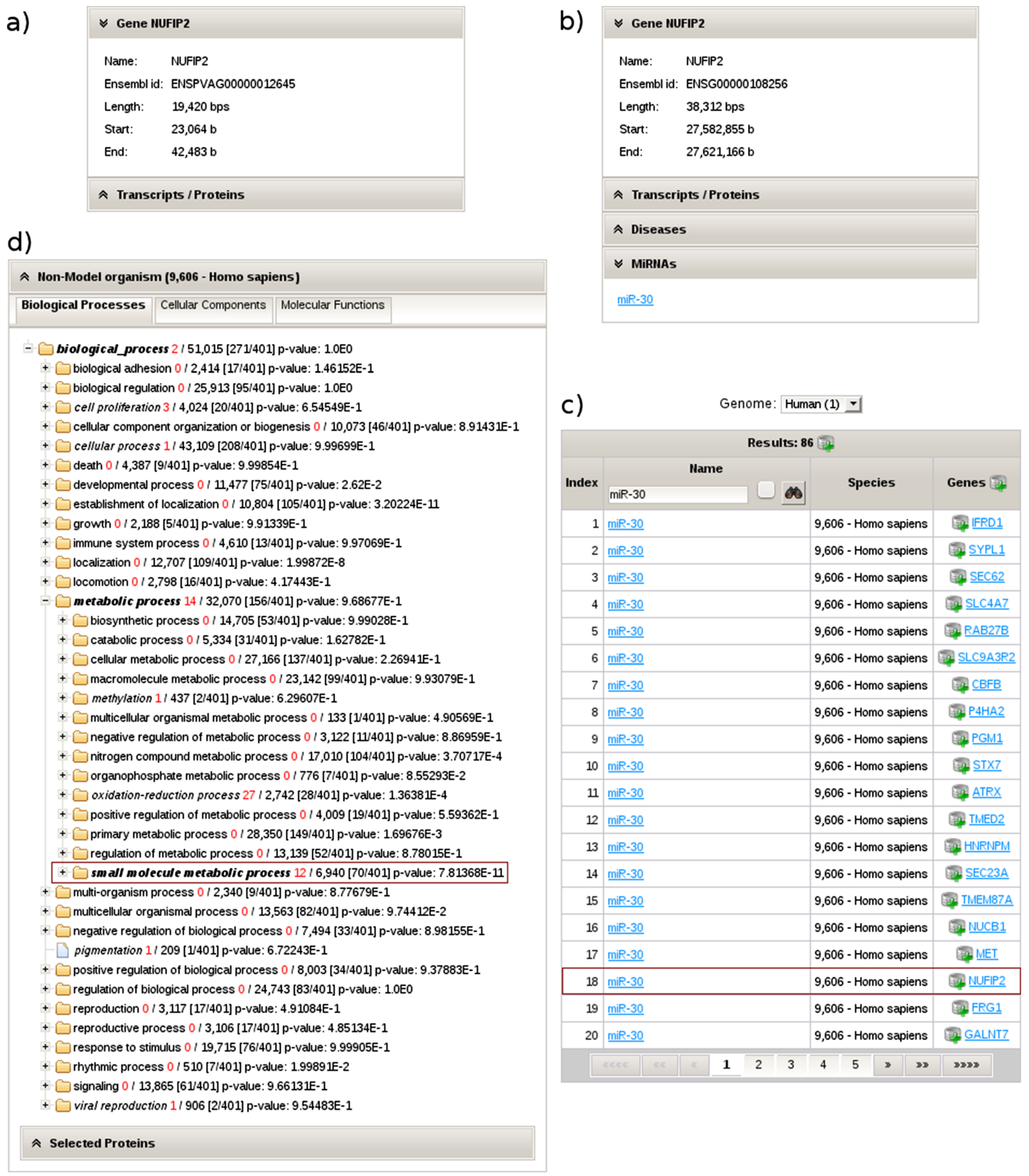
4.2.1.11
ENO1-001
ENO2-201
🟥 ENO3-001

Figure 2 Analyzing different molecular aspects of genes and proteins. a) Identification of disease genes – The user is interested in the disease Ectodermal dysplasia. From the returned list she focuses on autosomal recessive hypohidrotic ectodermal with three associated genes (red enclosed region). **b)** Interacting proteins – The graph shows the interaction network containing all genes which directly interact with the three genes in the snapshot. This interaction network comprises 58 genes. **c)** Identification of drug targets – 7 of the interacting genes are targets of known drugs. **d)** Integration into metabolic networks – One of the identified genes is involved in Glycolysis / Gluconeogenesis. Enzymes present in the snapshot are highlighted turquoise. Other enzymes present in the human genome are highlighted grey. Enzymes absent from the genome are not highlighted.

As an example, we integrated the genome of the large flying fox (*Pteropus vampyrus*) into ISAAC. Again, a researcher might start with a single bat gene of interest, e.g. *NUFIP2* (Figure 3a). To get a first glimpse of the function of this gene she can switch to a better understood organism like human (Figure 3b). Here, she can go to the microRNA module to check, whether this human ortholog of the bat gene is regulated by a microRNA. Indeed, she will find that the human *NUFIP2* gene is regulated by miR-30. The snapshot can now be enlarged by all other genes which are regulated by this microRNA, here 86 genes (Figure 3c). To see which functions are regulated by this miRNA, one can go to the GO enrichment module and search for overrepresented GO classifications (Figure 3d). In the example case, the user might focus on ‘small molecule metabolic process’ ($p = 7.8 \times 10^{-11}$). The researcher might switch the focus from miRNA to this defined function and add the genes to a new snapshot. An intersection with the previous snapshots enables to home in on genes which are regulated by miR-30 and involved in small molecule metabolism (20 genes). Finally, she translates this intersection back into the bat, resulting in 19 candidate genes which might be regulated by a microRNA and involved in small molecule metabolism. From here on, she could design an experiment to test, whether an ortholog of the human microRNA is indeed found in the bat and, if this is the case, whether the orthologous genes are indeed regulated by this microRNA.

Team work

Today, biological research is only rarely performed by a single lab on its own. In most cases, a lab works closely together with others to study different aspects of a gene using a wide range of techniques and organisms. The team working features of ISAAC might simplify the



a)

Gene NUFIP2

Name:	NUFIP2
Ensembl id:	ENSPVAG00000012645
Length:	19,420 bps
Start:	23,064 b
End:	42,483 b

Transcripts / Proteins

b)

Gene NUFIP2

Name:	NUFIP2
Ensembl id:	ENSG00000108256
Length:	38,312 bps
Start:	27,582,855 b
End:	27,621,166 b

Transcripts / Proteins

Diseases

MiRNAs

miR-30

c)

Genome: Human (1)

Results: 86

Index	Name miR-30	Species	Genes
1	miR-30	9,606 - Homo sapiens	IFRD1
2	miR-30	9,606 - Homo sapiens	SYPL1
3	miR-30	9,606 - Homo sapiens	SEC62
4	miR-30	9,606 - Homo sapiens	SLC4A7
5	miR-30	9,606 - Homo sapiens	RAB27B
6	miR-30	9,606 - Homo sapiens	SLC9A3R2
7	miR-30	9,606 - Homo sapiens	CBFB
8	miR-30	9,606 - Homo sapiens	P4HA2
9	miR-30	9,606 - Homo sapiens	PGM1
10	miR-30	9,606 - Homo sapiens	STX7
11	miR-30	9,606 - Homo sapiens	ATRX
12	miR-30	9,606 - Homo sapiens	TMED2
13	miR-30	9,606 - Homo sapiens	HNRNPM
14	miR-30	9,606 - Homo sapiens	SEC23A
15	miR-30	9,606 - Homo sapiens	TMEM87A
16	miR-30	9,606 - Homo sapiens	NUCB1
17	miR-30	9,606 - Homo sapiens	MET
18	miR-30	9,606 - Homo sapiens	NUFIP2
19	miR-30	9,606 - Homo sapiens	FRG1
20	miR-30	9,606 - Homo sapiens	GALNT7

d)

Non-Model organism (9,606 - Homo sapiens)

- ***biological_process*** 2 / 51,015 [271/401] p-value: 1.0E0
  - biological adhesion 0 / 2,414 [17/401] p-value: 1.46152E-1
  - biological regulation 0 / 25,913 [95/401] p-value: 1.0E0
  - *cell proliferation* 3 / 4,024 [20/401] p-value: 6.54549E-1
  - cellular component organization or biogenesis 0 / 10,073 [46/401] p-value: 8.91431E-1
  - *cellular process* 1 / 43,109 [208/401] p-value: 9.99699E-1
  - death 0 / 4,387 [9/401] p-value: 9.99854E-1
  - developmental process 0 / 11,477 [75/401] p-value: 2.62E-2
  - establishment of localization 0 / 10,804 [105/401] p-value: 3.20224E-11
  - growth 0 / 2,188 [5/401] p-value: 9.91339E-1
  - immune system process 0 / 4,610 [13/401] p-value: 9.97069E-1
  - localization 0 / 12,707 [109/401] p-value: 1.99872E-8
  - locomotion 0 / 2,798 [16/401] p-value: 4.17443E-1
  - ***metabolic process*** 14 / 32,070 [156/401] p-value: 9.68677E-1
    - biosynthetic process 0 / 14,705 [53/401] p-value: 9.99028E-1
    - catabolic process 0 / 5,334 [31/401] p-value: 1.62782E-1
    - cellular metabolic process 0 / 27,166 [137/401] p-value: 2.26941E-1
    - macromolecule metabolic process 0 / 23,142 [99/401] p-value: 9.93079E-1
    - *methylation* 1 / 437 [2/401] p-value: 6.29607E-1
    - multicellular organismal metabolic process 0 / 133 [1/401] p-value: 4.90569E-1
    - negative regulation of metabolic process 0 / 3,122 [11/401] p-value: 8.86959E-1
    - nitrogen compound metabolic process 0 / 17,010 [104/401] p-value: 3.70717E-4
    - organophosphate metabolic process 0 / 776 [7/401] p-value: 8.55293E-2
    - *oxidation-reduction process* 27 / 2,742 [28/401] p-value: 1.36381E-4
    - positive regulation of metabolic process 0 / 4,009 [19/401] p-value: 5.59362E-1
    - primary metabolic process 0 / 28,350 [149/401] p-value: 1.69676E-3
    - regulation of metabolic process 0 / 13,139 [52/401] p-value: 8.78015E-1
    - ***small molecule metabolic process*** 12 / 6,940 [70/401] p-value: 7.81368E-11
  - multi-organism process 0 / 2,340 [9/401] p-value: 8.77679E-1
  - multicellular organismal process 0 / 13,563 [82/401] p-value: 9.74412E-2
  - negative regulation of biological process 0 / 7,494 [33/401] p-value: 8.98155E-1
  - *pigmentation* 1 / 209 [1/401] p-value: 6.72243E-1
  - positive regulation of biological process 0 / 8,003 [34/401] p-value: 9.37883E-1
  - regulation of biological process 0 / 24,743 [83/401] p-value: 1.0E0
  - reproduction 0 / 3,117 [17/401] p-value: 4.91084E-1
  - reproductive process 0 / 3,106 [17/401] p-value: 4.85134E-1
  - response to stimulus 0 / 19,715 [76/401] p-value: 9.99905E-1
  - rhythmic process 0 / 510 [7/401] p-value: 1.99891E-2
  - signaling 0 / 13,865 [61/401] p-value: 9.66131E-1
  - *viral reproduction* 1 / 906 [2/401] p-value: 9.54483E-1

Selected Proteins

Figure 3 Working with non-model organisms. a) The user is interested in the bat gene *NUFIP2*. **b)** The panel displays the human ortholog and shows, that it is regulated by the microRNA miR-30. **c)** The table lists the 86 human genes regulated by the miRNA miR-30. **d)** 20 of the 86 genes regulated by miR-30 are overrepresented in the GO biological process classification "small molecule metabolic process" with a p-value: p = 7.81368x10^{-11}. 19 bat genes are orthologous to the 20 human genes regulated by the microRNA miR-30 and involved in the specific function "small molecule metabolic process", namely *ADPGK, ATP2A2, DOCK7, ELMOD2, GNAI2, GPD2, IDH1, JUN, MAT2A, NPR3, NT5C3, NT5E, P4HA2, PNP, PPP2R4, RAB27B, TMED2, UAP1* and *WNT5A*.

communication across different labs. As an example assume a consortium of groups working on actin nucleation (Figure 4). As a starting point, one researcher has deposited a manually curated list of human genes involved in actin nucleation. Within ISAAC a user group with all researchers of the consortium was created. Now the list can be published such that the whole group (and only this group) has access to the list. In addition to gene lists, also processes, i.e. results calculated within modules, can be published. For example, a user can publish an interaction network of the actin nucleation genes and all direct neighbors. A researcher of another group, working with a different species can now import the interaction network into her private pool. From here on she can use all features of ISAAC like mapping the genes into a new model species. Again, this result can be published, imported by another user who might manually curate this set and publish the results. Thus, the knowledge about gene sets of interest can be easily distributed within the consortium.

Conclusion

ISAAC enables non-computer trained researchers to explore gene lists under different biological aspects. From a user's point of view, the main difference to other related projects is that the gene lists can be changed interactively at any point of an analysis. Obviously this inherently carries the danger of losing track about how a

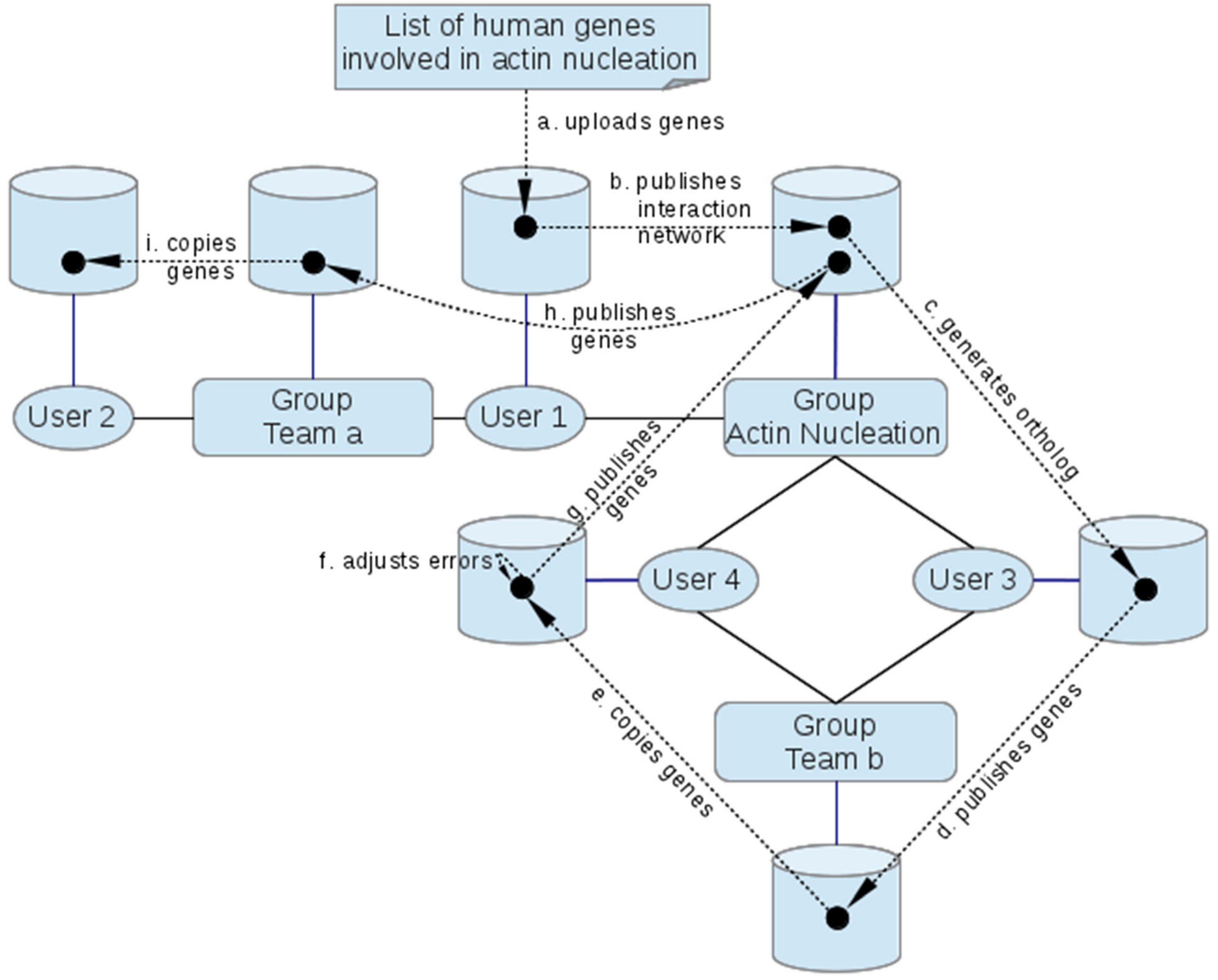


Figure 4 Working in teams. The diagram shows users and groups with their respective pools (blue lines). The users 1 and 2 are members of "Team a" with user 1 as coordinator. The users 3 and 4 belong to the group "Team b" and here user 3 is the coordinator. Furthermore, users 1, 3 and 4 are part of the group "Actin Nucleation" (users 1 and 4 are coordinators). The continuous black lines denote the memberships. Following the dashed arrows, user 1 imports into its private pool a manually curated list of human genes involved in actin nucleation **(a)** and publishes its protein interaction network to the group "Actin Nucleation" **(b)**. Using orthology mapping user 3 translates the interaction network into another species **(c)** and publishes it to the group "Team b" **(d)**. User 4 copies this list to his private pool **(e)**, manually curates it **(f)** and publishes this new list to the group "Actin Nucleation" **(g)**. The new list is published to group "Team a" by user 1 **(h)** and user 2 copies it to his private pool **(i)**.

set was generated. We therefore implemented an integrated version and logging system which supports the users on the persistence, administration and tracking of these sets. Together with the snapshot function, every part of the analysis can be traced and become a starting point for new analyses. Thus, ISAAC indeed enables the explorative mining for genes of interest. As new genomes are sequenced with an increasing pace, analyses crossing the species border become of increasing importance. As ISAAC includes information about orthologous relationships between genes, users can switch between species, automatically 'translating' gene sets from one species to another. Finally, ISAAC is not focused on single users. Instead, it offers options to share sets and even results of analyses between users and teams. Thus, not only a single user can look at a problem from different biological views, she can also let other researchers look at her genes to get an external view. Thereby, ISAAC supports multi team collaborative efforts getting ever more prominent in biological research.

From a programmer's point of view, ISAAC is based on an object oriented approach contrasting more workflow oriented programs, which are usually procedural. Sets of proteins, transcripts and genes with a well-defined structure together with comparison and operation methods build the core of this tool. Using this core, different modules can be implemented covering different biological aspects. The object oriented strategy and its modularity make this straightforward. Especially when performing highly explorative analyses, a user will need some breaks to e.g. gather further information. Therefore, there is no time out for a client web session. As long the web page is open, its session is held on the web server. To enable the integration of further modules, the source code is freely available from our web page.

Together with the web client, we developed an administration interface. Here, not only users and groups can be managed. More importantly, integration of third party data needed by a module can be carried out via the administration interface. This allows for example the straightforward addition of further genomes, as scripts which directly can insert Ensembl genomes are implemented and can be administrated via the web interface.

In summary, with its focus on small but highly explorative analyses ISAAC closes the gap between databases covering only on one or a few aspects of genes and proteins on the one hand and automated analysis tools which do not allow for interactive modifications of gene lists on the other.

Availability and requirements

Project name: ISAAC.
Project home page: http://isaac.bioapps.biozentrum.uni-wuerzburg.de.
Operating systems: Platform independent, tested on linux.
Web browser: Tested with Mozilla Firefox 16.0.2 and Internet Explorer 10.
Programming language: Java ≥ 1.7.
Other requirements: Java application server (Java EE 6 and JSF 2.0).
License: Free for academic users under the GNU Lesser General Public License (LGPL).

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

HB developed and implemented the system. JS designed and supervised the project. Both authors wrote, read and approved the final version of the manuscript.

Acknowledgments

HB was funded by DFG Grants SCHU2352/2-1 and the DFG Priority Programme SPP 1464: 'Principles and evolution of actin nucleator complexes'.

Received: 26 July 2013 Accepted: 10 January 2014
Published: 15 January 2014

References

1. Ouzounis CA: **Rise and demise of bioinformatics? Promise and progress.** *PLoS Comput Biol* 2012, **8:**e1002487.
2. Flicek P, Amode MR, Barrell D, Beal K, Brent S, Carvalho-Silva D, Clapham P, Coates G, Fairley S, Fitzgerald S, Gil L, Gordon L, Hendrix M, Hourlier T, Johnson N, Kähäri A, Keefe D, Keenan S, Kinsella R, Komorowska M, Koscielny G, Kulesha E, Larsson P, Longden I, McLaren W, Muffato M, Overduin B, Pignatelli M, Pritchard B, Riat HS: **Ensembl 2013.** *Nucleic Acids Res* 2012, **40:**D48–55.
3. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D: **The human genome browser at UCSC.** *Genome Res* 2002, **12:**996–1006.
4. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene Ontology: tool for the unification of biology.** *Nat Genet* 2000, **25:**25–29.
5. Amberger J, Bocchini CA, Scott AF, Hamosh A: **McKusick's Online Mendelian Inheritance in Man (OMIM®).** *Nucleic Acids Res* 2009, **37:**D798–796.
6. Knox C, Law V, Jewison T, Liu P, Ly S, Frolkis A, Pon A, Banco K, Mak C, Neveu V, Djoumbou Y, Eisner R, Guo AC, Wishart DS: **DrugBank 3.0: a comprehensive resource for "omics" research on drugs.** *Nucleic Acids Res* 2011, **39:**D1035–1041.
7. Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M: **KEGG: Kyoto Encyclopedia of Genes and Genomes.** *Nucleic Acids Res* 1999, **27:**29–34.
8. Szklarczyk D, Franceschini A, Kuhn M, Simonovic M, Roth A, Minguez P, Doerks T, Stark M, Muller J, Bork P, Jensen LJ, von Mering C: **The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored.** *Nucleic Acids Res* 2011, **39:**D561–568.
9. Kozomara A, Griffiths-Jones S: **miRBase: integrating microRNA annotation and deep-sequencing data.** *Nucleic Acids Res* 2011, **39:**D152–D157.
10. Dennis G, Sherman BT, Hosack DA, Yang J, Gao W, Lane HC, Lempicki RA: **DAVID: Database for Annotation, Visualization, and Integrated Discovery.** *Genome Biol* 2003, **4**.
11. Huang DW, Sherman BT, Lempicki RA: **Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.** *Nat Protoc* 2009, **4:**44–57.
12. Jiao X, Sherman BT, Huang DW, Stephens R, Baseler MW, Lane HC, Lempicki RA: **DAVID-WS: a stateful web service to facilitate gene/protein list analysis.** *Bioinformatics* 2012, **28:**1805–1806.
13. Weniger M, Engelmann JC, Schultz J: **Genome Expression Pathway Analysis Tool – Analysis and visualization of microarray gene expression data under genomic, proteomic and metabolic context.** *BMC Bioinformatics* 2007, **8:**179.
14. Khatri P, Draghici S, Ostermeier GC, Krawetz SA: **Profiling gene expression using Onto-Express.** *Genomics* 2002, **79:**266–270.

15. Khatri P, Voichita C, Kattan K, Ansari N, Khatri A, Georgescu C, Tarca AL, Draghici S: Onto-Tools: new additions and improvements in 2006. *Nucleic Acids Res* 2007, **35:**W206–211.
16. Kokocinski F, Delhomme N, Wrobel G, Hummerich L, Toedt G, Lichter P: FACT - a framework for the functional interpretation of high-throughput experiments. *BMC Bioinformatics* 2005, **6:**161.
17. Al-Shahrour F, Minguez P, Tárraga J, Montaner D, Alloza E, Vaquerizas JM, Conde L, Blaschke C, Vera J, Dopazo J: BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments. *Nucleic Acids Res* 2006, **34:**W472–476.
18. Al-Shahrour F, Minguez P, Tárraga J, Medina I, Alloza E, Montaner D, Dopazo J: FatiGO+: a functional profiling tool for genomic data. Integration of functional annotation, regulatory motifs and interaction data with microarray experiments. *Nucleic Acids Res* 2007, **35:**W91–W96.
19. Backes C, Keller A, Kuentzer J, Kneissl B, Comtesse N, Elnakady YA, Müller R, Meese E, Lenhof H-P: GeneTrail - advanced gene set enrichment analysis. *Nucleic Acids Res* 2007, **35:**W186–192.
20. Reimand J, Arak T, Vilo J: g:Profiler - a web server for functional interpretation of gene lists (2011 update). *Nucleic Acids Res* 2011, **39:**W307–W315.
21. Hu Z, Mellor J, Wu J, Yamada T, Holloway D, DeLisi C: VisANT: data-integrating visual framework for biological networks and modules. *Nucleic Acids Res* 2005, **33:**W352–357.
22. Croft D, O'Kelly G, Wu G, Haw R, Gillespie M, Matthews L, Caudy M, Garapati P, Gopinath G, Jassal B, Jupe S, Kalatskaya I, Mahajan S, May B, Ndegwa N, Schmidt E, Shamovsky V, Yung C, Birney E, Hermjakob H, D'Eustachio P, Stein L: Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res* 2011, **39:**D691–697.
23. Doniger SW, Salomonis N, Dahlquist KD, Vranizan K, Lawlor SC, Conklin BR: MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biol* 2003, **4:**R7.
24. Masseroli M, Martucci D, Pinciroli F: GFINDer: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining. *Nucleic Acids Res* 2004, **32:**W293–300.
25. Sealfon RS, Hibbs MA, Huttenhower C, Myers CL, Troyanskaya OG: GOLEM: an interactive graph-based gene-ontology navigation and analysis tool. *BMC Bioinformatics* 2006, **7:**443.
26. Huang DW, Sherman BT, Lempicki RA: Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res* 2009, **37:**1–13.
27. Ingenuity System. [http://www.ingenuity.com]
28. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Amin DR, Schwikowski B, Ideker T: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003, **13:**2498–2504.
29. Zhang B, Kirov S, Snoddy J: WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res* 2005, **33:**W741–748.
30. Wang J, Duncan D, Shi Z, Zhang B: WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013. *Nucleic Acids Res* 2013, **41:**W77–W83.
31. Glez-Peña D, Gómez-López G, Pisano DG, Fdez-Riverola F: WhichGenes: a web-based tool for gathering, building, storing and exporting gene sets with application in gene set enrichment analysis. *Nucleic Acids Res* 2009, **37:**W329–W334.
32. Kinsella RJ, Kähäri A, Haider S, Zamora J, Proctor G, Spudich G, Almeida-King J, Staines D, Derwent P, Kerhornou A, Kersey P, Flicek P: Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database (Oxford)* 2011, **2011**. bar030.
33. Hunter S, Jones P, Mitchell A, Apweiler R, Attwood TK, Bateman A, Bernard T, Binns D, Bork P, Burge S, de Castro E, Coggill P, Corbett M, Das U, Daugherty L, Duquenne L, Finn RD, Fraser M, Gough J, Haft D, Hulo N, Kahn D, Kelly E, Letunic I, Lonsdale D, Lopez R, Madera M, Maslen J, McAnulla C, McDowall J: InterPro in 2011: new developments in the family and domain prediction database. *Nucleic Acids Res* 2011, **40:**D306–D312.
34. Kersey PJ, Staines DM, Lawson D, Kulesha E, Derwent P, Humphrey JC, Hughes DST, Keenan S, Kerhornou A, Koscielny G, Langridge N, McDowall MD, Megy K, Maheswari U, Nuhn M, Paulini M, Pedro H, Toneva I, Wilson D, Yates A, Birney E: Ensembl Genomes: an integrative resource for genome-scale data from non-vertebrate species. *Nucleic Acids Res* 2012, **40:**D91–D97.
35. The UniProt Consortium: Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res* 2012, **40:**D71–D75.
36. Grossmann S, Bauer S, Robinson PN, Vingron M: Improved detection of overrepresentation of Gene-Ontology annotations with parent child analysis. *Bioinformatics* 2007, **23:**3024–3031.
37. Bauer S, Grossmann S, Vingron M, Robinson PN: Ontologizer 2.0 - a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics* 2008, **24:**1650–1651.
38. Papadopoulos GL, Reczko M, Simossis VA, Sethupathy P, Hatzigeorgiou AG: The database of experimentally supported targets: a functional update of TarBase. *Nucleic Acids Res* 2009, **37:**D155–D158.

doi:10.1186/1471-2105-15-18
Cite this article as: Baier and Schultz: ISAAC - InterSpecies Analysing Application using Containers. *BMC Bioinformatics* 2014 **15**:18.