



Descripción de Interfaces de Instancias EGA Comunidad



IMPACT

Infraestructura de Medicina de Precisión
asociada a la Ciencia y la Tecnología

Program	IMPACT: Infraestructura de Medicina de Precisión asociada a la Ciencia y la Tecnología		
Project Name	IMPACT-Data: Programa de Ciencia de Datos de IMPACT		
Expedient	IMP/00019		
Duration	January 2021 – December 2023		
Work Package	WP3 – Genomics		
Task	T3.1 - Implementación de sistemas de almacenamiento de información genómica primaria mediante instancias de local EGAs		
Deliverable	E3.2. Descripción de Interfaces de Instancias EGA Comunidad		
Version	1.1.1		
Due Date	30/06/2022	Approval Date	17/05/2023
Responsible	CRG		
Dissemination Level	X	PU	Public
		CO-IMP	Confidential, only IMPACT pillars members, including the evaluation commission from IMPACT.
		CO-DATA	Confidential, only IMPACT-Data members, including the evaluation commission from IMPACT.

<i>Authors</i>		
<i>Organization</i>	<i>Name</i>	<i>Role</i>
BSC-CNS	Lidia López	Coordination
EGA-CRG	Jordi Rambla	Author
EGA-CRG	Amy Curwin	Author
BU-ISCIII	Isabel Cuesta	Reviewer
IIS-FJD	Pablo Minguez	Reviewer

<i>Versions History</i>			
<i>N.</i>	<i>Date</i>	<i>Description</i>	<i>Author</i>
v 0.0	04/05/2022	Created	A. Curwin (EGA-CRG)
v 0.1	09/05/2022	Outline of sections	A. Curwin (EGA-CRG)
v 0.9	23/06/2022	Content filled and sent to reviewers	J. Rambla/A. Curwin (EGA-CRG)
v 1.0	30/06/2022	Final version with comments addressed	J. Rambla/A. Curwin (EGA-CRG)
v 1.1	17/05/2023	Visibility changed to public and approved	Comité Directivo
v 1.1.1	14/06/2023	Cambio de formato para publicar en la Web de IMPaCT	David Velasco (ISCIII)

Content

Content	4
Figures	5
Executive Summary	6
Introduction	7
Audience	7
Topic	7
Relation to other Deliverables	7
Deliverable Structure	7
1. Overview of The EGA Community	8
1.1 The LocalEGA solution	9
2 Input interfaces	10
3 Output interfaces	11
3.1 Accessing sections of a file	12
3.2 EGA-Quickview	12
4 Discovery	13
4.1 Beacon v2	13
5 Authentication and authorization interfaces	15
5.1 OpenID connect	15
5.2 ssh Authentication	15
6 Conclusions	16
References	17
Acronyms and Abbreviators	17

Figures

Figure 1: The Data Management, Discovery and Sharing components (eg. input and output interfaces)8

Figure 2: A diagram of the LocalEGA components and the relationships among them (indicated with lines and arrows). The different colors depict the 3 security zones in a LocalEGA instance.9

Figure 3: A schematic representation of how Beacon works. (A) Beacon API implementation and (B) A Beacon query and aggregated response 13

Executive Summary

The European Genome-phenome Archive (EGA) is a service for law-compliant storing and sharing of all types of human genomic data and associated metadata. The EGA is currently a centralized service co-managed by the EMBL-EBI and the CRG. In 2022, several countries and the Central EGA institutions (CRG and EMBL-EBI) have established the Federated EGA. The Federated EGA (FEGA) is a network to support data management requirements inside different jurisdictions. Federated EGA nodes offer EGA services to researchers within their jurisdiction. However, Community EGA nodes are individual institutions or initiatives with human genetic and genomic data intended to be shared with the research community. Similarly, IMPaCT-Data nodes, that could eventually be part or not of the EGA Federation, would be organized in a federated network and, hence, the technical problems and suggested solutions are almost identical. This document builds on E3.1 “Requisitos de un Nodo Local EGA”, and describes in more detail the interfaces developed by the EGA that could be adopted by an EGA Community or IMPaCT-Data node to facilitate management and sharing of sensitive human data.

Introduction

Audience

This deliverable is envisioned as a useful document for those institutes who would like to establish a Community instance of EGA or IMPaCT-Data node (the “**Node**” from this point on). It describes user authentication, input and output interfaces, as well Beacon v2, a discovery interface. We provide background information as well and links to technical documentation to enable implementation of the interfaces described within.

Topic

The mission of IMPaCT-Data project is to set the basis for the successful set up of the Spanish personalized medicine program. This deliverable builds on previous deliverables to describe tools that can enable entities such as hospitals, research centres, etc. to participate in the management, analysis and sharing of genomic data (and associated metadata). This data is the basis for the development of personalized medicine and therefore, empowering the entities of our countries with the tools, knowledge and network to manage this type of data, is instrumental to the final mission of the IMPaCT-Data project. The current model is for Nodes to minimize moving sensitive data from the Node, while public metadata have no restrictions. Files will be stored encrypted in the Nodes located at different institutions, while public metadata goes to Central EGA.

Relation to other Deliverables

This is the second deliverable of WP3 and builds on E3.1 “Requisitos de un Nodo Local EGA” (delivered in December 2021), that described the requirements for setting up a Local EGA Community node. E3.2 also relates to E5.4. “Requisitos Técnicos Puesta en Marcha Sistemas Beacon v. 1.0” (delivered in December 2021) that described technical requirements for Beacon. Finally E3.2 is also related to E3.3 “Informe Final de un Nodo EGA” (to be delivered 2023).

Deliverable Structure

This deliverable describes EGA Community Nodes interfaces. The structure is as follows: Section 1. Overview of the EGA Community; Section 2. Input interfaces (2.1. sFTP for ingestion); Section 3. Output interfaces (3.1. Accessing sections of a file and 3.2. EGA-Quickview (sFTP); Section 4. Discovery (4.1. Beacon v2); Section 5. Authentication and authorization interfaces (5.1. OpenID Connect and 5.2. ssh Authentication).

1. Overview of The EGA Community

The concept of an EGA Community was described in E3.1. To briefly review, the vision for EGA Community is that they host a set of valuable data to which they want to provide controlled access, but the remit or structure of the projects does not allow them to engage deeply in the EGA Federation. Potential EGA Communities include a variety of projects and organizations, therefore, there is also a heterogeneity of capacities in terms of technical solutions for data managing and sharing. The solutions range from having no infrastructure at all to having a fully functional and mature solution in place. IMPaCT-Data partners follow a very similar scenario and, therefore, solutions described here would apply to them too.

The integration of EGA nodes in the Federated EGA network is based on an agreed set of programmatic interfaces (APIs) that allows for internal node independence for some services (e.g. data storage, data access or data distribution) while integrating seamlessly into the network for other services (e.g. data discovery, accessioning). All of them in the context of a proper management of user identities and permissions (AAI).

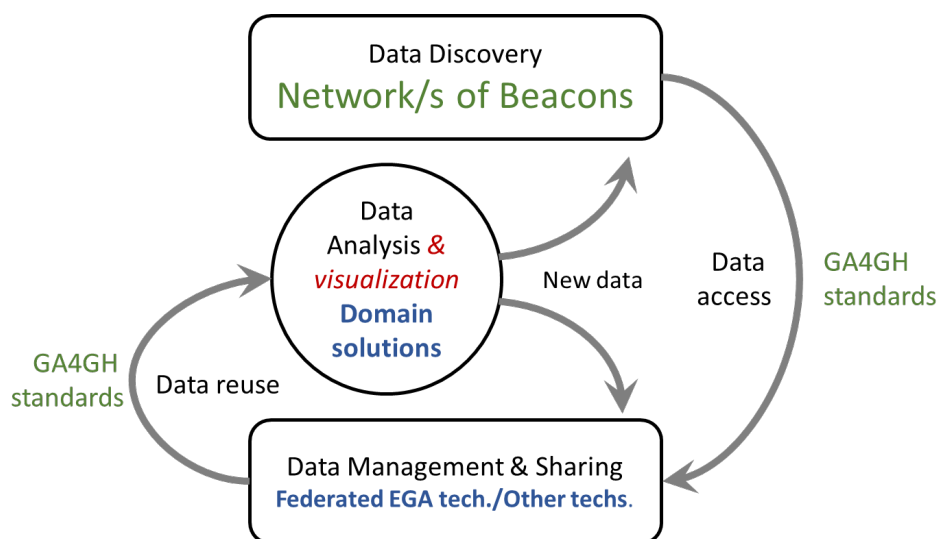


Figure 1: The Data Management, Discovery and Sharing components (eg. input and output interfaces)

The EGA team has developed, or participated in the development of, several products and technologies that serve as the foundation for the processes described¹. In particular:

- Input and output interfaces for data management and sharing.
- The GA4GH Beacon² for data discovery.

¹ Freeberg et.al. The European Genome-Phenome Archive in 2021. *Nucleic Acids Research*, Volume 50, Issue D1, 7 January 2022, Pages D980–D987, <https://doi.org/10.1093/nar/gkab1059>

² <https://beacon-project.io/>

- Authentication and authorization integration

1.1 The LocalEGA solution

In this document we will refer to the “integrated solution” that implements the interfaces and features we recommend for the data management part of the platform.

The integrated solution includes the following components:

- An inbox
- A long-term database and file storage³
- An ingestion pipeline

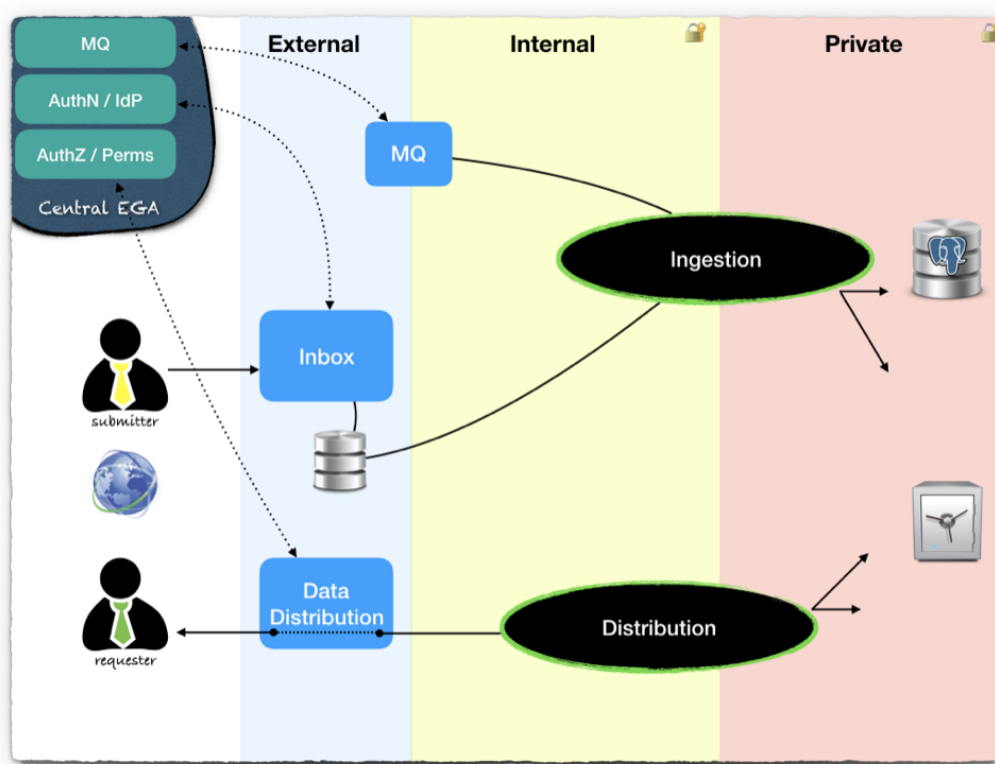


Figure 2: A diagram of the LocalEGA components and the relationships among them (indicated with lines and arrows). The different colors depict the 3 security zones in a LocalEGA instance.


The current model is for Nodes to minimize moving sensitive data from the Node, while public metadata have no restrictions. Files will be stored encrypted in the Nodes located at different institutions, while public metadata goes to Central EGA.

³ The institution must provide the storage space. The latter is connected and made available to the software (as mountpoints, for example). This goes for the backup storage, and the database as well. The database software is provided, but it is the responsibility of the institution to choose where the underlying data reside (as well as back it up regularly/if necessary).

In short, users upload encrypted files into a Local EGA inbox, located in the relevant Node. The ingestion pipeline moves the encrypted files from the inbox into the long-term storage, and saves information about the files in the database. In the process, each ingested file obtains an Accession ID, which identifies it uniquely across the EGA. The distribution system allows requesters to securely access the encrypted files in the long-term storage, using the accession id, if permissions are granted by a Data Access Committee (DAC).

When a user, authorized by the corresponding DAC, wants to access the files in a given dataset, he or she needs to login into the distribution system with the provided credentials and leverage the tools available, like EGA Quickview (more information in a section below).

The communication between the Node and the Central EGA is performed via a secure messaging protocol (AMQP) that is resilient to connection failures.



Detailed information on:
LocalEGA
could be found here:
LocalEGA [documentation](#) & [repository](#)

2 Input interfaces

According to the process described above, the first step in setting up a repository of genomic files is to establish a data storage solution and a mechanism for uploading or moving the files to that storage.

Input interfaces are responsible for allowing the data to land in the data storage. All sensitive data in that storage must be encrypted in a safe and efficient way. The Global Alliance for Genomics and Health (GA4GH) has designed an encrypted file format, Crypt4GH⁴, that uses strong and modern encryption algorithms and a format that is adequate for huge files, like the genomics ones are.

If, instead of leveraging the LocalEGA solution, a Node wants to implement a compatible input solution on its own, the required steps are:


1. Encrypt the file/s to upload using the crypt4gh format: the file content needs to be encrypted by the submitter using the repository public key
2. The user authenticates into the file upload solution by using one of the following technologies
 1. for HTTP: OpenID Connect
 2. for sFTP: credentials or ssh-key
3. The user uploads the encrypted file/s

⁴ Alexander Senf, Robert Davies, Frédéric Haziza, John Marshall, Juan Troncoso-Pastoriza, Oliver Hofmann, Thomas M. Keane, Crypt4GH: a file format standard enabling native access to encrypted data, *Bioinformatics*, Volume 37, Issue 17, 1 September 2021, Pages 2753–2754, <https://doi.org/10.1093/bioinformatics/btab087>

The uploaded file stays in the inbox until a process picks it, validates it and moves it to the permanent safe and secure storage. Usually, this is triggered upon reception of the metadata that completes the submission. In the case of a Node, this process could be internal (where no external submitter is involved) and the trigger could happen at the most convenient point for the Node as the process is internal to the system. This process is called **file ingestion**.

According to the previous description, any solution that is able to generate and upload files in crypt4gh format would be valid as an input solution. For example, one bash script that encrypts the file using existing tools and then uploads it using an sFTP command-line would be an acceptable solution.

However, no integrated solution, that is receiving a file and encrypting it on the fly, is available outside the one developed by the EGA team, and this is the one being suggested for IMPaCT-Data.



Detailed information on:
crypt4gh
could be found here:
[crypt4gh specification](#) & [paper](#)

3 Output interfaces

Similarly to the ingestion operation, the **distribution** part is also based on a combination of available technologies:

- a storage component, e.g., a POSIX or S3 filesystem
- an optional file transfer server, e.g., sFTP or Aspera
- a decryption tool for crypt4gh files
- a script that links all steps together

Again, no integrated solution exists outside of the developments introduced in the current document. The EGA has developed one that aims for maximum security while being transparent to the user and, more importantly, to the tools that the user plans to use on the files.

The distribution process is more complex than the ingestion one as it involves two additional steps: 1) checking that the user is authorized to access the requested dataset (and hence its files) and 2) after that verification, encrypting the file exclusively for that specific user.

Therefore, the interfaces required for distribution are:


1. One file transfer solution (e.g., sFTP, Aspera)
2. One for authenticating the user

1. HTTP with OpenID Connect
2. sFTP with credentials or ssh-key
3. One for retrieving the file access permissions granted to the authenticated user
4. One for re-encrypting the file exclusively for the requesting user
 1. (recommended option) by the user providing its public key
 2. (alternative option) by the system generating a key pair and securely sending it to the user
5. Once the users have the files on their system, a decryption utility for accessing the file contents

Similarly to the ingestion use case, any in-house solution/s that includes the above steps would be acceptable as it is transparent to the overall IMPaCT-Data operations.

3.1 Accessing sections of a file

The approach of the solution described above is to download the whole file. However, there are use cases where the user is only interested in a section of a file (e.g., a given chromosome or a set of regions inside the genome). For such cases, the recommendation would be an integrated solution like EGA-Quickview (see below) or an `htsget` service:



Detailed information on:
htsget
could be found here:
[specification](#), [documentation](#) & [paper](#)

Integration of `htsget` with the authentication and authorization depends on the specific `htsget` implementation used and is a key aspect to consider in this scenario.

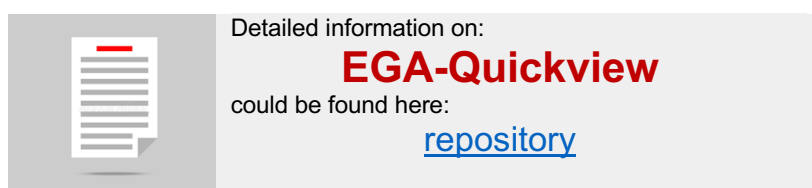
3.2 EGA-Quickview

EGA-QuickView is an output interface tool that allows fast and easy queries of a given dataset or file without the need of downloading it entirely. For example, if a user wants to know if mutation X exists in a particular location, they can query only that region of genomic data, find the answer they need rapidly without taking the time to download everything.

[EGA-QuickView](#) is a FUSE (remote) file system using a combination of [sshfs](#) and [crypt4ghfs](#). It allows communication with the EGA distribution servers over ssh, to download files (chunk by chunk) in Crypt4GH format and decrypt them transparently.

It is useful for a user to quickly browse through a file, but loses its purpose if the user plans on scanning the entire file. For that latter, it is more appropriate to download the files using the recommendations at the beginning of the Output interfaces section.

One key aspect of using EGA-Quickview is that contents are secure for the user posting the request by using its public key (hence the user must decrypt the content by using her/his private key). The recommendation is that the key is stored inside the user profile in the authentication and authorization component as this makes the process much more transparent to the user.



4 Discovery

4.1 Beacon v2

One of the main challenges of clinical data discoverability is the lack of tools for the secure and federated discovery of identifiable human patients' data without jeopardizing the privacy or ownership of such datasets. Currently, most of the molecular and genomics data generated in hospitals are not utilized for further research at all.

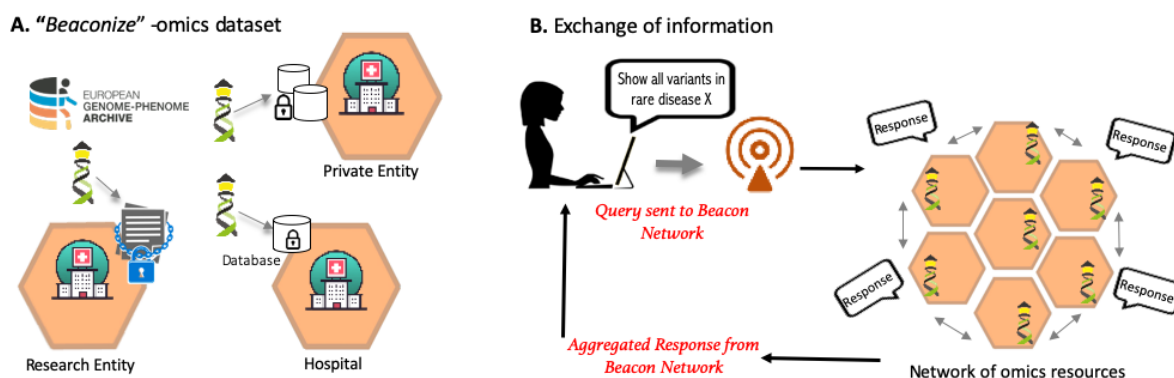


Figure 3: A schematic representation of how Beacon works. (A) Beacon API implementation and (B) A Beacon query and aggregated response

EGA, along with ELIXIR and GA4GH, has been involved to develop tools and techniques to tackle this challenge. Beacon API aims to alleviate the problem of ethical clinical data sharing by enabling the federated discovery of genomic variants, clinical data, and other associated information, all while maintaining the privacy of the dataset. Beacon v2 API

provides several Beacon *flavours*, or different Beacon usage scenarios, to query the data, i.e., a **boolean** yes/no responses or **counts** of matched entries (e.g., number of genomic variants) or **full** beacon responses. This way, a beacon implementor will always be in full control of their dataset and can choose how much or how little information they want to expose.

For example, a beacon implementor, in this case a clinician working with rare diseases, chooses to share their finding i.e., rare mutations diagnosed in their patients. This clinician can implement the beacon on their dataset and choose the type of response (beacon *flavour*) as '**aggregated**' response type for public users. When a public user makes a query on the Beacon Network looking for a particular rare mutation that has already been diagnosed in above clinic, the beacon response will return that this mutation has already been reported in NN patients (i.e., an aggregated count response) at clinic C, and a handover link will be provided to connect to the clinic directly. Therefore, in this scenario a) the data did not move from the clinic b) the data became part of the Beacon Network, and therefore discoverable, and c) the response did not expose any information except the counts of the result matched.

After a successful trial of previous Beacon versions, the Beacon community submitted the Beacon version 2 (Beacon v2) protocol for the GA4GH approval in the Spring 2022. Beacon v2 is a major extension of its previous Beacon versions where the Beacon community from around the world gathered weekly to push this new version farther in terms of data discovery, keeping the original goal intact i.e., federated discovery of identifiable genomics data with tight privacy controls.

Beacon v2 provides several upgrades from its previous versions:

- More informative queries, like filtering by gender or age, cohort discovery etc.
- An option to trigger the next step in the data access process, e.g., who to contact or which are the data use conditions.
- An option to jump to another system where the data could be accessed, e.g., if the Beacon is for internal use of the hospital, to provide the ID of the EHR of the patients having the mutation of interest.
- Annotations about the variants found, among which the expert/clinician conclusion about the pathogenicity of a given mutation in a given individual or its role in producing a given phenotype.
- Information about cohorts.



Detailed information on:

Beacon v2

could be found here:

[documentation](#) - [implementation](#)

5 Authentication and authorization interfaces

An Authentication and Authorization management solution (AuthN/AuthZ) is central to Node operations. Authentication is verifying the identity of a user, while authorisation is confirming a user has access rights to specific information. The ability to manage and audit who has access to what is required for preventing malicious or accidental unauthorised data access. The EGA Community Node AuthN/AuthZ implementation must follow the GA4GH AuthN/AuthZ recommendations, ensuring that data access can be managed interoperably with other GA4GH AuthN/AuthZ-compatible resources.

Users can interact with multiple services that a Node has built on top of AuthN/AuthZ as described in the previous sections, namely:

- HTTP services that rely on OpenID Connect
- ssh login dependent services like: sFTP or Aspera

5.1 OpenID connect

Any Node service that is provided over HTTP must support authentication based on, and compatible with, the OpenID Connect protocol. This is the GA4GH and also ELIXIR recommendation and it is implemented by popular solutions like Keycloak.

There are also servers that provide identity services (IdP or Identity Provider) like LifeSciences AAI which allows a user to authenticate and use the provided token to access the desired service.

The user will register at the identity provider, where he/she should provide some profile information and some security elements, like a mechanism for retrieving forgotten passwords or security questions... the provided credentials would be used for authentication at that service. Moreover, the LS AAI handles the important bit about vouching for the user's identity, e.g., by verifying the user belongs to the said institution.

5.2 ssh Authentication


On the other hand, ssh authentication uses an Operating System native mechanism and is available for non-HTTP based services.

The ssh authentication is available using a username+password combination or based on a public-private key pair. The latter is the recommended way.

The user can use a key in the ED25519 format, as it could be used both for ssh authentication and for crypt4gh encryption. The server only is provided with the public part.

Any recent openssh implementation (6.7+) is able to manage the ED25519 key format.

The Node is responsible for establishing a mechanism for the users to submit their public keys and for storing it in a way that could be easily retrieved by the services that require them.



Detailed documentation on:
AAI
could be found here:
[passports](#)

6 Conclusions

In conclusion, this document briefly describes the various interfaces of an EGA Community Node including references to further documentation where a Node can find details about how to implement the different components of the solution.

References

1. Freeberg et.al. The European Genome-Phenome Archive in 2021. *Nucleic Acids Research*, Volume 50, Issue D1, 7 January 2022, Pages D980–D987, <https://doi.org/10.1093/nar/gkab1059>
2. <https://beacon-project.io/>
3. Alexander Senf, Robert Davies, Frédéric Haziza, John Marshall, Juan Troncoso-Pastoriza, Oliver Hofmann, Thomas M. Keane, Crypt4GH: a file format standard enabling native access to encrypted data, *Bioinformatics*, Volume 37, Issue 17, 1 September 2021, Pages 2753–2754, <https://doi.org/10.1093/bioinformatics/btab087>

Acronyms and Abbreviators

In the following table there are some acronyms and abbreviators used in the deliverable.

CEGA	Central European Genome-phenome Archive
CRG	Centre for Genomic Regulation
DAC	Data access committee
EBI	European bioinformatics institute
EGA	European Genome-phenome Archive
EMBL	European molecular biology laboratories
FEGA	Federated European Genome-phenome Archive
GA4GH	Global alliance for genomics and health
IMPACT	Infraestructura de Medicina de Precisión asociada a la Ciencia y la Tecnología (Spanish initials)
IMPACT-Data	Data program for IMPACT
WP	Work Package