INFORMATICS

Human Mutation

OFFICIAL JOURNAL
HGVS
HUMAN GENOME VARIATION SOCIETY
www.hgvs.org

The Matchmaker Exchange API: Automating Patient Matching Through the Exchange of Structured Phenotypic and Genotypic Profiles

Orion J. Buske,[1,2,3]* François Schiettecatte,[4] Benjamin Hutton,[5] Sergiu Dumitriu,[3] Andriy Misyura,[3] Lijia Huang,[6] Taila Hartley,[6] Marta Girdea,[2,3] Nara Sobreira,[7] Chris Mungall,[8] and Michael Brudno[1,2,3]

[1]Genetics and Genome Biology Program, The Hospital for Sick Children, Toronto, Canada; [2]Department of Computer Science, University of Toronto, Toronto, Canada; [3]Centre for Computational Medicine, The Hospital for Sick Children, Toronto, Canada; [4]FS Consulting LLC, Salem, Massachusetts; [5]Wellcome Trust Sanger Institute, Cambridge, UK; [6]Children's Hospital of Eastern Ontario Research Institute, Ottawa, Ontario, Canada; [7]McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, Maryland; [8]Genomics Division, Lawrence Berkeley National Laboratory, Berkeley, California

For the Matchmaker Exchange Special Issue

Received 28 April 2015; accepted revised manuscript 24 July 2015.
Published online in Wiley Online Library (www.wiley.com/humanmutation). DOI: 10.1002/humu.22850

ABSTRACT: **Despite the increasing prevalence of clinical sequencing, the difficulty of identifying additional affected families is a key obstacle to solving many rare diseases. There may only be a handful of similar patients worldwide, and their data may be stored in diverse clinical and research databases. Computational methods are necessary to enable finding similar patients across the growing number of patient repositories and registries. We present the Matchmaker Exchange Application Programming Interface (MME API), a protocol and data format for exchanging phenotype and genotype profiles to enable matchmaking among patient databases, facilitate the identification of additional cohorts, and increase the rate with which rare diseases can be researched and diagnosed. We designed the API to be straightforward and flexible in order to simplify its adoption on a large number of data types and workflows. We also provide a public test data set, curated from the literature, to facilitate implementation of the API and development of new matching algorithms. The initial version of the API has been successfully implemented by three members of the Matchmaker Exchange and was immediately able to reproduce previously identified matches and generate several new leads currently being validated. The API is available at https://github.com/ga4gh/mme-apis.**

Hum Mutat 36:922–927, 2015. © 2015 Wiley Periodicals, Inc.

KEY WORDS: **patient matchmaking; genomic API; rare disease; GA4GH; HPO; Matchmaker Exchange**

Introduction

Rare genetic disorders collectively affect around 350 million people worldwide, but the number of people affected by any one of these disorders can be extremely small. These individuals may be seen by different clinicians and sequenced at different centers, with each individual's data being stored in one of a rapidly growing number of different databases and patient registries. Siloing of data severely impedes the discovery of genetic causes of these disorders, while directly copying such data across various resources is impossible due to a number of legal and privacy concerns. Developing efforts such as the Global Alliance for Genomics and Health (GA4GH) APIs are designed to facilitate the exchange of genetic data between such databases; however, these are currently targeting genetic data and hypothesis-driven queries. To address the need for flexible data sharing amongst resources with rare disease patient data, we developed the Matchmaker Exchange Application Program Interface (MME API), a data format and protocol for querying databases to identify individuals with similar phenotypic profiles and genetic variation, a process we call "matchmaking."

The MME API specifies the format of both the query, which is sent to participating databases (which we call "matchmaker services"), and the response, which contains information about matching individuals in the remote database. The initial version of this API follows a *query-by-example* philosophy, in which the request is simply a description of the individual to be matched and the response is a list of the descriptions of similar individuals. Because the API is built around the description of an individual rather than a complex query language, it is easy to understand, straightforward to implement, and provides the various databases the flexibility of experimenting with matching algorithms and regulating the amount of data that is disclosed. Further, because the case is used as the query, more specific and complete case records will return more relevant matches, thus encouraging users to submit the most complete and specific case information possible.

The sharing and automated analysis of genetic and phenotypic data has necessitated standardization using a number of ontologies and controlled terminologies. In this API, we use the Sequence Ontology (Eilbeck et al., 2005) to describe the class of the genetic variants (e.g., whether it is insertion, deletion, or SNV; missense or stopgain, etc.) and the Human Phenotype Ontology (HPO) (Köhler et al., 2014) to describe patient phenotypes. The HPO has over 11,000 terms corresponding to phenotypic abnormalities, which are structured from general (e.g., "abnormality of the nervous system") to specific (e.g., "atonic seizures"). Importantly, the HPO has the "true path rule", which states that the presence of a lower-level term implies the presence of all ancestors of the term

*Correspondence to: api@matchmakerexchange.org

Contract grant sponsor: National Human Genome Research Institute, NIH (1U54HG006542).

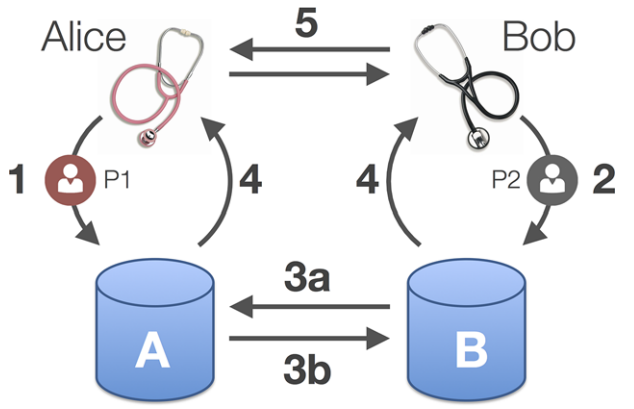

Figure 1. Overview of the matchmaking process, in which (1) Alice deposits case P1 into Matchmaker A; (2) sometime later, Bob deposits a similar case P2 into Matchmaker B; (3a) Matchmaker B then sends a match request with a description of P2 to Matchmaker A and (3b) receives a match response with a description of similar patients (including P1) from Matchmaker A; (4) Matchmaker A informs Alice and Matchmaker B informs Bob of the P1-P2 match; and (5) Alice and Bob communicate if the match warrants further investigation.

(a patient with "atonic seizures", by definition, also has "seizures" and an "abnormality of the nervous system"). This feature makes it possible to "obfuscate" a term by using one of its ancestors instead, and to match distinct but related terms by identifying shared ancestors.

Many MME partners perform some form of internal matchmaking to identify similar patients within their database, but each organization has a different focus, collects different types of data, and stores their data in different formats. The MME API provides a standardized language for exchanging patient profiles in order to enable matchmaking between patient databases. Here, we present a description of the MME API, the method used to authenticate endpoints of this API within the MME, and a test dataset available to verify that endpoints are behaving as expected and assist in the development of novel matching algorithms. The API has been developed in collaboration with the GA4GH and uses standard field names and data formats wherever possible. It complies with current best practices for Web APIs and uses Javascript Object Notation (JSON) to encode all content that is sent and received.

Methods and Results

The Matchmaker Exchange (MME) API

The matchmaking workflow

An overview of the match request and response process is shown in Figure 1. The user starts by contributing a case to one of the Matchmaker Exchange services (Philippakis et al., 2015, this issue). On behalf of the user, the matchmaker service then queries other MME services using the MME API. These other services use the structured patient data in the query to identify and return descriptions of similar cases within their respective databases. They are not permitted to store request data for uses other than analytics and diagnostics (i.e., the data exchanged over the API does not become a part of the data stored by the receiving services). Similar cases found through the API are then reported to the users for evaluation. The users can then follow up with each other on any promising matches using contact information provided with the query and response. It is currently up to each MME service to define the process for alerting their respective users of the match (i.e., step 4 in Fig. 1).

Format

The API defines a set of data types, each with a corresponding set of properties (e.g., the Disorder type has two properties, "id", which is mandatory, and "label", which is optional). An object is a particular example (instantiation) of a type (an example Disorder object in JSON format is: {"id": "OMIM:269880", "label": "SHORT syndrome"}). The core of the format is a specification of an individual with relevant phenotypic and/or genotypic features (the Patient type, defined in Table 1). A match request (see Fig. 2B) contains a single case in this format, used as the query, and the match response contains a scored list of the most similar cases in the remote system, also in this format. The patient type is designed to be flexible to facilitate matchmaking between cases with varying degrees of phenotypic and/or genotypic detail. It can contain a list of diagnoses, phenotypic features, and/or genotypic features, along with metadata such as an identifier, sex, and contact information of the submitter of the case (so that promising matches can be followed up on). There are few required fields, making it easy to implement regardless of the data stored by the matchmaker service, and many optional fields, enabling additional information to be conveyed to improve the accuracy of matchmaking and help users interpret the matches.

Standardized identifiers and ontologies are used wherever possible. Diagnoses are specified using OMIM (Hamosh et al., 2005) or Orphanet (http://www.orphadata.org/) identifiers. Each phenotypic feature (a Feature object) is specified using a term from the HPO, and can be recorded as either observed (the default) or explicitly absent (it may be important for similarity measures and differential diagnosis to know if particular features or co-morbidities were explicitly checked for but not observed in the individual). To protect privacy, phenotypic features can be intentionally obfuscated in the query or the response by substituting HPO terms with ancestors of those terms. Each genotypic feature (a GenomicFeature object) represents a candidate gene or variant believed to be directly involved in the individual's phenotype. It contains a gene identifier, specified as an HGNC gene symbol, an Ensembl gene identifier, or an Entrez gene identifier, and can include details about the type of variant (specified as a Sequence Ontology term) and/or the specific variant with respect to a reference genome. Extensive documentation is available on the GitHub page (https://github.com/ga4gh/mme-apis).

The match response (see Fig. 2D and Table 1) contains a list of the cases in the database most similar to the case specified in the query, scored according to the particular matchmaker service's matching algorithm. Scores must be a number between 0.0 (a poor match) and 1.0 (an excellent match), but scores are not yet comparable across matchmaker services as matching algorithms vary. Currently, only an overall score for the strength of each match is required, but more detailed scoring of the phenotypic and genotypic aspects of each match will likely be added in future versions.

API versioning

The MME API is semantically versioned (http://semver.org/), with version numbers taking the form "X.Y", where X is incremented for major releases and Y is incremented for backwards-compatible minor releases. Every request must specify the API version within

Table 1. Fields of the MME API

Type	Property	Req*	Expected type	Description	Example
Match request	patient	√	Patient	Query patient	*See Fig. 2B lines 2–53 and Patient type*
Patient	id	√	String	Unique, persistent patient identifier	"F0000011"
	label		String	Human-readable identifier, *no personally identifiable information*	"174_170258"
	contact	√	Contact	Contact details for depositor of patient record	*See Fig. 2B, lines 5–9 and Contact type*
	species		String	NCBI taxon identifier	"NCBITaxon:9606"
	sex		String	Genetic sex ("FEMALE", "MALE", "OTHER")	"FEMALE"
	ageOfOnset		String	Age interval at onset of the majority of the symptoms (HPO term identifier)	"HP:0003623"
	inheritanceMode		String	Mode of inheritance (HPO term identifier)	"HP:0000006"
	disorders		List of Disorders	List of diagnoses	*See Fig. 2B, lines 12–17 and Disorder type*
	features	†	List of Features	List of phenotypic traits	*See Fig. 2B, lines 18–33 and Feature type*
	genomicFeatures	†	List of GenomicFeatures	List of candidate causal genes and variants	*See Fig. 2B, lines 34–52 and GenomicFeatures type*
Contact	name	√	String	Name of the clinician or organization	"Kym Boycott"
	institution		String	Institution of the clinician	"FORGE Canada"
	href	√	String	Contact URL; either public Web page or email address (mailto)	"http://dx.doi.org/10.1016/j.ajhg.2011.12.001"
Disorder	id	√	String	OMIM or ORDO identifier	"MIM:136140"
	label			Human-readable description	"Floating-Harbor Syndrome"
Feature	id	√	String	HPO term identifier	"HP:0004322"
	label		String	Human-readable description	"Short stature"
	observed		String	The feature has been *explicitly observed* ("yes") or *explicitly not observed* ("no")	"Yes"
	ageOfOnset		String	Age interval at onset (HPO term identifier)	"HP:0003577"
GenomicFeature	gene	√	Gene	Candidate gene	*See Fig. 2B, lines 36–38 and Gene type*
	variant		Variant	Candidate variant in gene	*See Fig. 2B, lines 39–45 and Variant type*
	zygosity		Number	Allelic dosage (1: heterozygous, 2: homozygous)	1
	type		GenomicFeatureType	cDNA effect of the mutation	*See Fig. 2B, lines 47–50 and GenomicFeatureType type*
Gene	id	√	String	Gene symbol, Ensembl gene ID, or Entrez gene ID	"SRCAP"
Variant	assembly	√	String	Reference assembly identifier	"GRCh37"
	referenceName	√	String	Chromosome	"16"
	start	√	Number	Start position (0-based)	30748691
	end		Number	End position (0-based, exclusive)	30748692
	referenceBases		String	VCF-style reference allele of at least one base	"C"
	alternateBases		String	VCF-style alternate allele of at least one base	"T"
GenomicFeatureType	id	√	String	SO term identifier	"SO:0001587"
	label		String	Human-readable description	"STOPGAIN"
MatchResponse	results	√	List of MatchResults	List of similar/matching patients	*See Fig. 2D, lines 2–10 and MatchResults type*
MatchResult	score	√	Match Score	Scoring details for the match	*See Fig. 2D, lines 4–6 and MatchScore type*
	patient	√	Patient	Matching patient	*See Fig. 2D, line 7 and Patient type*
MatchScore	patient	√	Number	Overall match score (in the range [0, 1], where 0.0 is a poor match and 1.0 is a perfect match)	0.983

Example values from a patient description in Hood et al. (2012).
*The "Req" column contains a check mark for properties that are mandatory for objects of the given class.
†It is preferred to have both the "features" and "genomicFeatures" properties defined for every Patient object; it is mandatory to have at least one of the two.

the HTTP Accept header, and the remote server must provide the API version of the response in the Content-Type header of every response (see Fig. 2A and C).

Error handling

The remote server should use HTTP status codes to report any error encountered processing the match request. Table 2 contains a list of status codes and their meanings with regards to this API. The error response should include a JSON-formatted body with a human-readable "message" containing further details about the error (see Fig. 2E). The exact error message is up to the implementer, and additional fields can be provided with further information.

Request Authentication in the Matchmaker Exchange

All communication between servers in the Matchmaker Exchange must occur over secure HTTP (HTTPS), and requests are currently authenticated through a simple yet effective protocol. If Matchmaker B wishes to accept match requests from Matchmaker A, Matchmaker B first securely sends a secret authentication token to Matchmaker A (e.g., through encrypted email). We recommend the authentication token be a randomly generated SHA1 hexadecimal digest. This

A

```
POST /baseURL/match HTTP/1.1
Host: b.org
Accept: application/vnd.ga4gh.matchmaker.v1.0+json
X-Auth-Token: 854a439d278df4283bf5498ab020336cdc416a7d
```

B

```
{
  "patient": {
    "id": "F0000011",
    "label": "174_170258",
    "contact": {
      "name": "Kym Boycott",
      "institution": "FORGE Canada",
      "href": "http://dx.doi.org/
        10.1016/j.ajhg.2011.12.001"
    },
    "sex": "FEMALE",
    "inheritanceMode": "HP:0000006",
    "disorders": [
      {
        "id": "MIM:136140",
        "label": "Floating-Harbor syndrome"
      }
    ],
    "features": [
      {
        "id": "HP:0004322",
        "label": "Short stature"
      },
      {
        "id": "HP:0000878",
        "label": "11 pairs of ribs"
      },
      {
        "id": "HP:0000369",
        "observed": "no",
        "label": "Low-set ears"
      },
      ...
    ],
    "genomicFeatures": [
      {
        "gene": {
          "id": "SRCAP"
        },
        "variant": {
          "assembly": "GRCh37",
          "referenceName": "16",
          "start": 30748691,
          "referenceBases": "C",
          "alternateBases": "T"
        },
        "zygosity": 1,
        "type": {
          "id": "SO:0001587",
          "label": "STOPGAIN"
        }
      }
    ]
  }
```

C

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ga4gh.matchmaker.v1.1+json; charset=UTF-8
```

D

```
{
  "results": [
    {
      "score": {
        "patient": 0.94283
      },
      "patient": {...}
    },
    ...
  ]
}
```

E

```
HTTP/1.1 406 Not Acceptable
Content-Type: application/vnd.ga4gh.matchmaker.v2.2+json; charset=UTF-8
```

```
{
  "message": "unsupported API version",
  "supportedVersions": [ "2.0", "2.1", "2.2" ]
}
```

Figure 2. An example match request and response, based on a patient description in Hood et al. (2012). **A**: The HTTP header of the POST request to a matchmaker service at b.org, serving the API from baseURL. The Accept header specifies that the response should conform to version 1.0 of the MME API. The X-Auth-Token header is set to the secret token that b.org provided the querier to authenticate match requests. **B**: An example request body, describing a particular patient with Floating-Harbor Syndrome (additional features omitted for brevity). **C**: The HTTP header of a successful matchmaking response, indicated by the 200 OK status code. The Content-Type header specifies that the response conforms to version 1.1 of the MME API, which is backwards compatible with the version 1.0 query. **D**: An example response body, containing a list of matching cases and corresponding match scores (patient details and additional matches omitted for brevity). **E**: The HTTP header and body of a failed matchmaking response, in which the server does not support the API version of the query (version 1.0) and responds with an appropriate message, a Content-Type containing the latest API version supported by the server, and a list of all supported API versions (optional).

Table 2. HTTP status codes and their intended use within the MME API

HTTP status code	Reason phrase	Description
200	OK	No error
400	Bad request	Missing/invalid data
401	Unauthorized	Missing/invalid authentication token
405	Method not allowed	Invalid method (POST required)
406	Not acceptable	Missing/unsupported API version
415	Unsupported media type	Missing/invalid content type
422	Unprocessable entity	Missing/invalid request body
500	Internal server error	Default error

authentication token must be specified as the X-Auth-Token header of all requests that Matchmaker A makes to Matchmaker B (see Fig. 2A). Matchmaker B will then verify the authentication token and may perform additional checks such as validating the originating IP address of the request (though this is not required). We are currently exploring support for a federated user authentication scheme, such as OAuth 2.0 (http://oauth.net/), in future versions of the API.

Test Data

In order to facilitate testing the ability of systems to query, match, and respond to requests, we have compiled a standardized test dataset of 50 de-identified individuals spanning 22 disorders. These cases were selected from publications by the FORGE Canada (Beaulieu et al., 2014) and Care4Rare Canada projects

(http://care4rare.ca/), and deliberately include conditions with diverse phenotypes. Some of the conditions involve multiple organ systems (e.g., OMIM:269880 SHORT syndrome; OMIM:182212 Shprintzen-Goldberg Syndrome), whereas others mainly affect a single system (e.g., OMIM:614665 Meconium ileus; OMIM:243150 Intestinal atresia, multiple). In addition, multiple individuals with variable severity were included for many of the disorders (e.g., OMIM:615960 Cerebellar Dysplasia and Cysts; OMIM:615273 Congenital disorder of glycosylation, type IV), which serve as internal controls for evaluating the performance of matchmaking algorithms. These test cases are available in the MME API JSON format, and are annotated with phenotypic features, the diagnosed disorder (OMIM identifier), and the causal variant(s). New matchmaking organizations can use this dataset internally, to verify that the query and response are formatted correctly and the matching is accurate, or externally, to verify that links to other matchmaker services are functioning properly. In these cases, an additional property of the Patient object, "test", should be set to true. This informs the system being queried that the query is a test, allowing it to respond accordingly. Normally, the system being queried will match against real patient data, return any matches, and notify users of identified matches. With a test query, the system should run the match against test data, return any matches, and suppress any notifications.

Deployment of the API across the MME Network

The MME API is currently implemented at the DECIPHER (Chatzimichali et al., 2015, this issue), GeneMatcher (Sobreira et al., 2015, this issue), and PhenomeCentral (Buske et al., 2015, this issue) portals. We have validated the API through two means. First, through the use of the test data (described above), which recovered all of the expected matches. Second, as a preliminary test with clinical cases, we used the MME API to find matches for unsolved PhenomeCentral cases within GeneMatcher. We identified 60 unsolved PhenomeCentral cases submitted by the Care4Rare Canada project, which together included 45 different candidate genes (1–5 candidate genes per record). At least one match was found for 37 out of 60 PhenomeCentral cases, with 33 matching cases returned in total. Of the 33 matches, 16 were duplicate records (entered by the same clinician in both systems) and 2 were excluded because GeneMatcher had many ($\geq$30) candidate genes per record. We followed up on the 10 matching genes within the remaining 15 matching records, with 6 of the gene matches classified as false positives (i.e., phenotypes of the two patients were not significantly similar after clinician review), 2 of the gene matches still unresolved, and 2 of the gene matches classified as potentially significant hits with additional validation currently underway. GeneMatcher currently matches only on gene since most of the cases do not have phenotypic information, which may contribute to the false positive rate of this test.

Discussion

The Matchmaker Exchange is an international collaboration to facilitate the exchange of phenotypic and genotypic data for cases of rare disorders. The MME API presented here was designed to enable automated sharing of this data between multiple patient databases. The overarching principle guiding the design was to create a framework that is flexible enough to support a large number of data types and workflows, as the various members of the Matchmaker Exchange support varying depth of phenotypic and genetic data. The details of the algorithms used in each matchmaker service are also still in development. We decided on a hypothesis-free approach, in which the patient record defines the query and the receiving site determines how to optimally process the query, as it likely has the best understanding of the data available and how to use it to measure patient similarity. One added advantage of this approach is that to obtain optimum matches, the query patient has to be deeply phenotyped, thus encouraging contribution of data into the network. We believe that our approach will have utility beyond the rare disease community, and have contributed our APIs to the Global Alliance for Genomics and Health. Wherever possible, we coordinated field names and data formats with those used by the GA4GH APIs, and will continue to engage in the development of these standards.

While this API has proven successful for the first iteration of matchmaking, we are also considering extensions that should improve the efficacy of the API. These include improvements to the security/privacy configurations and a gradual adoption of hypothesis-driven queries. We believe that two changes could enhance the privacy protections offered by the MME API. First, some MME sites currently obfuscate the provided data before returning it, and require direct communication between the submitting users before showing full patient data. Currently, the API does not support reporting when data has been obfuscated; however, this information may be useful for the receiving user. Secondly, a centralized identification framework, using a technology such as OpenID, would enable users to have a single sign-on for all of the MME partners, as well as allowing the receiving site to make decisions on what data to show in response to a query based on the user's profile and their membership in the receiving site.

Finally, we expect the current hypothesis-free nature of the API to develop into a partially hypothesis-driven approach. Toward this end the API should allow for weighing or requiring of features (e.g., specifying a specific gene or phenotype as "required", suggesting a scoring function to be applied when computing a match score, or filtering the results based on a feature). In our tests, we have found increasing need for such features, as the scoring schemes differ significantly between matchmaker services, making expected results difficult to validate.

Acknowledgments

We are grateful to all member of the Matchmaker Exchange working group for steering our effort, as well as to the leadership of the International Rare Disease Research Consortium (IRDiRC), the Global Alliance for Genomics and Health (GA4GH), and the Clinical Genome Resource (ClinGen) for supporting the MME project. The development of the MME API was supported by funding from the National Human Genome Research Institute (1U54HG006542) as well as Genome Canada and the Canadian Institutes for Health Research through the Large Scale Advanced Research (LSARP) and Bioinformatics/Computational Biology (BCB) Programs. OB was supported by the Garron Family Cancer Centre and Hospital for Sick Children Foundation Student Scholarship Program.

Disclosure statement: The authors have no competing interests to declare.

References

Beaulieu CL, Majewski J, Schwartzentruber J, Samuels ME, Fernandez BA, Bernier FP, Brudno M, Knoppers B, Marcadier J, Dyment D, Adam S, Bulman DE, et al. 2014. FORGE Canada Consortium: outcomes of a 2-year national rare-disease gene-discovery project. Am J Hum Genet 94:809–817.

Buske OJ, Girdea M, Dumitriu S, Gallinger B, Hartley T, Trang H, Misyura A, Friedman T, Beaulieu C, Bone WP, Links AE, Washington NL, et al. 2015. PhenomeCentral:

a portal for phenotypic and genotypic matchmaking of patients with rare genetic diseases. Hum Mutat 36:931–940.

Chatzimichali EA, Brent S, Hutton B, Perrett D, Wright CF, Bevan AP, Hurles ME, Firth HV, Swaminathan GJ. 2015. Facilitating collaboration in rare genetic disorders through effective matchmaking in DECIPHER. Hum Mutat 36:941–949.

Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M. 2005. The Sequence Ontology: a tool for the unification of genome annotations. Genome Biol 6:R44.

Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res 33:D514–D517.

Hood RL, Lines MA, Nikkel SM, Schwartzentruber J, Beaulieu C, Nowaczyk MJ, Allanson J, Kim CA, Wieczorek D, Moilanen JS, Lacombe D, Gillessen-Kaesbach G, et al. 2012. Mutations in SRCAP, encoding SNF2-related CREBBP activator protein, cause Floating-Harbor syndrome. Am J Hum Genet 90:308–313.

Köhler S, Doelken SC, Mungall CJ, Bauer S, Firth HV, Bailleul-Forestier I, Black GCM, Brown DL, Brudno M, Campbell J, FitzPatrick DR, et al. 2014. The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data. Nucleic Acids Res 42:D966–D974.

Philippakis A, Azzariti D, Beltran S, Brookes A, Brownstein C, Brudno M, Brunner H, Buske O, Carey K, Doll C, Dumitriu S, Dyke S, et al. 2015. The matchmaker exchange: A platform for rare disease gene discovery. Hum Mutat 36:915–921.

Sobreira N, Schiettecatte F, Valle D, Hamosh A. 2015. GeneMatcher: A matching tool for connecting investigators with an interest in the same gene. Hum Mutat 36:928–930.