# PatientMatcher: a customizable Python-based open-source tool for matching undiagnosed rare disease patients via the MatchMaker Exchange network

Chiara Rasi[1], Daniel Nilsson[2], Måns Magnusson[2], Nicole Lesko[3], Kristina Lagerstedt-Robinson[4], Anna Wedell[2], Anna Lindstrand[4], Valtteri Wirta[1], and Henrik Stranneheim[1]

[1]Science for Life Laboratory
[2]Karolinska Institute
[3]Karolinska Universitetssjukhuset
[4]Karolinska University Hospital

February 22, 2024

**Abstract**

The amount of data available from genomic medicine has revolutionized the approach to identify the determinants underlying many rare diseases. The task of confirming a genotype-phenotype causality for a patient affected with a rare genetic disease is often challenging. In this context, the establishment of the MatchMaker Exchange (MME) network has assumed a pivotal role in bridging heterogeneous patient information stored on different medical and research servers. MME has made it possible to solve rare disease cases by "matching" the genotypic and phenotypic characteristics of a patient of interest with patient data available at other clinical facilities participating in the network. Here, we present PatientMatcher (https://github.com/Clinical-Genomics/patientMatcher), an open-source Python and MongoDB-based software solution developed by Clinical Genomics facility at the Science for Life Laboratory in Stockholm. PatientMatcher is designed as a standalone MME server, but can easily communicate via REST API with external applications managing genetic analyses and patient data. The MME node is being implemented in clinical production in collaboration with the Genomic Medicine Center Karolinska at the Karolinska University Hospital. PatientMatcher is written to implement the MME API and provides several customizable settings, including a custom-fit similarity score algorithm and adjustable matching results notifications.

## Introduction

The increasing accessibility of accurate genomic data via next-generation sequencing (NGS) (Shendure and Ji, 2008; L. Metzker, 2010) has opened new avenues to a cost-effective diagnosis of the genetic determinants underlying many rare diseases (RDs) (M. Boycott et al., 2013). Obtaining a molecular diagnosis for a patient with a rare disease constitutes often a challenging task; currently typically less than 50% of patients receive a molecular diagnosis despite strong suspicion of an underlying genetic determinant (E. Soden et al., 2014; Lee et al., 2014; Yang et al., 2014; Stranneheim et al., 2021). However, a powerful approach for disease gene discovery is through identification of other patients with a similar phenotype; by establishing a cohort of similar patients, the likelihood of identifying the shared genomic determinant is strongly increased. The establishment of the MatchMaker Exchange (MME)(A. Philippakis et al., 2015) federated network has dramatically improved the process of "matchmaking" patients across clinical laboratories' and research

centers' databases(R. Azzariti and Hamosh, 2020). MME APIs(J. Buske et al., 2015) simplify sharing of selected patient data with the purpose of identifying cases with shared phenotypes and genotypic profiles. An MME matching event results in a notification sent to the patients' data submitters belonging to the two participating centers, so they can evaluate the matching features and eventually establish a gene or variant-disease causality for the given patient features. The obvious advantage of this tool is that single users utilizing the service do not need to worry about different database standards and data formats, as MME nodes communicate via standardized protocols and return results in a common and language-independent data format (JSON).

The Clinical Genomics facility at Science for Life Laboratory (SciLifeLab) Stockholm has been collaborating with the regional healthcare at Karolinska University Hospital to provide whole genome sequencing (WGS) - based rare disease (RD) diagnostics since 2015. Through this collaboration, termed Genomic Medicine Center Karolinska(Stranneheim et al., 2021), to date more than 6000 RD patients corresponding to more than 10,000 samples (6000 at the time PatientMatcher was launched) have been analyzed making this the largest clinical whole genome sequencing effort in Sweden. This collaboration is responsible for the genetic testing of the vast majority of RD cases in the Stockholm region, accounting for ˜2500 samples sequenced annually. Clinical Genomics is additionally a founding member of Genomic Medicine Sweden (https://www.genomicmedicine.se) and the Nordic Alliance for Clinical Genomics (http://www.nordicclinicalgenomics.org). In this framework, PatientMatcher was developed by Clinical Genomics as a clinical diagnostic decision support tool to aid clinicians and researchers at partner institutes solving RD cases. PatientMatcher is now being implemented at the Genomic Medicine Center Karolinska in clinical production to establish a controlled, fully integrated data sharing capability as part of the diagnostic workflow.

## Design and implementation

At the time this software was developed, there already existed several open-source solutions which could be adopted by a patient database owner in order to connect to the MME network as an independent node (https://github.com/ga4gh/mme-apis/wiki/Implementations). After an analysis of the existing implementations, we came to the conclusion that none of them really addressed our needs, as these implementations were too tightly coupled to data structures and routines of a specific host research center, or software demos not intended for production settings (https://github.com/MatchmakerExchange/reference-server). The first technical reason that prompted us to launch PatientMatcher was the need to develop an application written in Python (https://www.python.org/). This is the language of choice for most of the projects developed at our facility and for this reason the project will have better chances of being maintained over time. Another obvious advantage is that developing the solution in a very popular programming language, will likely increase the chances that PatientMatcher or some of its modules will be used by other research centers or diagnostic laboratories willing to connect to MME as distinct nodes. The second technical challenge that led us to develop a custom solution, was the necessity of storing data in a documented-oriented database such as MongoDB (https://www.mongodb.com/), where patient data documents are very similar to data objects used in Scout (https://github.com/Clinical-Genomics/scout), the application used by our clinical laboratories for handling results from NGS analyses. Additionally, MongoDB saves documents in JSON, the same format used by MME nodes for exchanging patient data via HTTP requests. Technical considerations aside, our primary reason to develop the software from scratch was the opportunity to introduce a highly customizable patient similarity scoring algorithm, to help data contributors to fine-tune the parameters of interest to be used in the patient similarity computation. PatientMatcher consists of a Python (3.6+) backend connected to a web app built in Flask 2.0+ (https://flask.palletsprojects.com/en/2.0.x/). The application data is stored in a MongoDB database.

The program backend contains the command to update database resources: HPO and disease term ontologies, respectively downloaded from the OBO Foundry (https://github.com/OBOFoundry) and the Jenkins automation server from the Monarch Institute (https://ci.monarchinitiative.org/). These resources are the

core of the software's phenotype similarity score algorithm. The command line is additionally used to add or remove MME clients (connected nodes allowed to run queries on PatientMatcher by exhibiting a security token that is unique for each node) and MME nodes (external nodes queried by PatientMatcher using a token assigned in turn by these servers).

PatientMatcher is basically a Representation State Transfer (REST) API tool that allows to programmatically submit data, download results and perform exhaustive comparison against the internal database data set or submit queries to external nodes. The application is written to implement the Matchmaker Exchange API specifications. (J. Buske et al., 2015) The available server endpoints are illustrated in Table 1.

# Matching algorithm

When the server receives a matching request from an external node (external matching) or from a user wishing to match a specific patient against all other patients on the server (internal matching), the query triggers a matching algorithm, which computes the similarity between the query patient and all patients stored in the database. As for other MME implementations and per MME API specifications, patient similarity is measured by a similarity score between 0 (no matching features) and 1 (exact matching of all patient's features). The maximum number of patients returned by the server is a parameter which can be customized by editing the "MAX_RESULTS" field in the app settings. The default value for this parameter is 5. Patient matches are returned in order of descending similarity with the query patient (i.e., high similarity matches are presented first in the list of results). Similarity score computation in PatientMatcher is taking into account genomic similarity and phenotype similarity across patients. The weight of these factors is numerically evaluated into a GTScore and a PhenoScore, where the sum of these two contributes to the total similarity score (result score) between query and matched patient. The relative importance of GTScore and PhenoScore in the computation can be customized by the server administrator by modifying the values of the parameters named "MAX_GT_SCORE" and "MAX_PHENO_SCORE" in the app configuration settings. The default value for both these parameters is 0.5, meaning an equivalent impact of phenotype and genotype similarity on the result score. This design was made to address diverse requirements from different data contributors. For example, a clinical laboratory might be storing patient genetic information with little availability of diagnoses or phenotype terms. In that case it makes sense to set the weight of the phenotype matching to zero and rely on genotype matching only. On the other hand, country regulations might not allow sharing of accurate genetic information, for instance variant details, but only gene symbols. If detailed patient diagnoses are also available for these patients, using both GTScore and PhenoScore when running the similarity algorithm will increase the chances of producing meaningful matches.

### 2.1.1 Genotype Matching Algorithm

When the parameter MAX_GT_SCORE is set to a valuer higher than zero and the query data contains genotype features (gene or variant information), a genotype similarity score will be evaluated between query patient and every patient (matched patient) contained in the database. All patients matching at least one of the candidate genes present in the query will be initially selected as matches. As specified in the MME API, candidate genes should preferably be described by an Ensembl ID (i.e., "ENSG00000101680"), but it is possible to search the database using patients with genes represented by HGNC symbols (i.e., "*LAMA1* ") and Entrez IDs (i.e., "6481"). The algorithm is designed to assign higher matching scores to patients described by fewer genotype features. For instance, a query patient described by a unique gene (A) that matches a database patient described by the same gene (A) will produce a higher genotype score than a query patient described by two genes (A and B). Genotype score (GT_SCORE) is quantified by the formula:

GT_SCORE = MAX_GT_SCORE / [?]fs

This number is calculated by dividing the MAX_GT_SCORE by the sum of the feature scores (fs) measured from the matching of each genotype feature of query patient against a matching patient. For example,

according to this definition, assuming a MAX_GT_SCORE of 0.5, each gene from a patient described by 3 genes will have a fs of a third of 0.5 (0.1666). If a gene from the query patient does not match any gene of the matched patient, then the fs for that feature would have a value of 0. In the eventuality of exact matching of gene and gene variant, the fs would be assigned with the highest possible value for the feature (0.1666). Incomplete gene matches (gene matching and no variant matching or no variant metadata available for the provided genes) are assigned with an arbitrary value or a quarter of the fs for the feature (0.1666/4). By calculating the GT_SCORE in this manner, the algorithm produces an accurate numerical estimate of the similarity between all genotype features of matching patients. This, in turn, allows the server to return patient hits sorted by descending genetic similarity with the query patient and not simply all patients that match any of its genes.

PatientMatcher also provides the possibility to evaluate and assign scores to matching variants that are outside genes. Feature scores from variant matching outside genes are assigned with the same fs as exact (gene + variant) matchings. It is worth mentioning that the genotype matching algorithm contains a liftover functionality that allows to quantify the similarity between patients containing genomic features described in different genome builds.

### 2.1.2 Phenotype Matching Algorithm

PatientMatcher is calculating phenotype matching scores based on both patient features and disorders. Patient features are described by HPO terms (Köhler et al., 2014) provided for query and matched patients, while disorders are represented by Decipher (V. Firth et al., 2009), OMIM (Hamosh et al., 2000) or Orphanet (Pavan et al., 2017) entries. If patients to be compared contain features and disorders, these descriptors will be both accounted for and each of them will contribute to half of the resulting phenotype score (PHENO_SCORE). Similarity between HPO features will be solely considered in the computation when disorders are not provided for one or both patients. Whereas disease terms comparison in the algorithm is still relatively unpolished (it consists in a pairwise comparison of diagnoses between the patients), semantic similarity metrics between HPO terms and their ancestor terms are calculated as simGIC measures(Pesquita, 2007; Pesquita et al., 2008). The original algorithm used for creating the phenotype ontology and comparing the patients in PatientMatcher is available in the Patient-Similarity package (https://github.com/buske/patient-similarity). Since the HPO is curating resources bridging disease terms with their associated HPO entries, we envision that in future software releases disease similarity comparisons will be also calculated as semantic relationships between terms.

## 2.2 Email notifications

Email notifications can be enabled by administrators via specific parameters present in the software configuration file. In order to modulate the amount of information included in the email notification body and thereby limit the extent of potentially sensitive information distributed via email, there exists two notification options: 1) complete notifications containing the entire description of matching patients (including gene names, variants and phenotypes), and 2) partial notifications report with only patient IDs and patient's clinician's contact information. Email notifications are sent to the patient contact only in case of positive matches from requests triggered by the same user, by another user within PatientMatcher (internal matches) or an external user from an MME connected node (external matches).

# Integration with Scout data at Genomic Medicine Center Karolinska, Stockholm

PatientMatcher was developed as a standalone software with the aim of providing an easy-to-administer MME server for any research institute or clinical laboratory wanting to pursue a connection to the MME

network as an independent node. The application does not have a graphical user interface (GUI) and patient data entry is achieved by handling incoming HTTP POST requests containing authentication tokens as well as patient data information. The instance of PatientMatcher hosted at GMCK contains an integration with Scout, the browser-based decision support software platform used to display and analyze WGS analyses from RD cases. These cases include mostly patients from the three collaborating clinical diagnostic laboratories at the Karolinska University Hospital: The Center for Inherited Metabolic Diseases, the Department of Clinical Genetics and the Department of Immunology and Transfusion Medicine. These patients, analyzed either as singletons, trios or larger family structures, present symptoms such as intellectual disabilities, inborn errors of metabolism, mitochondrial and neuromuscular diseases, primary immune deficiencies as well as connective tissues and skeletal diseases, among other disorders(Stranneheim et al., 2021). In the current setup Scout and PatientMatcher are distinct software instances residing on a single server, but depending on local IT infrastructure they can, if needed, be installed on different servers as they communicate via REST APIs. On the Scout portal, the MME integration feature is visible by all users. Access to the functionality is however granted to designated users authorized to submit cases to the MME network. A typical interpretation of a clinical case using Scout involves reviewing variants available for the affected individual(s) of a case with the goal of identifying one or a few candidates responsible for a specific phenotype. Phenotypes in Scout can be assigned at the case and the individual level as a list of HPO terms and/or OMIM diagnoses (https://omim.org/). The requirements to submit a case to PatientMatcher is that the Scout case should have up to three variants "pinned" as possible causatives and/or its phenotype should be described (by HPO and/or OMIM terms). It is noteworthy that since Scout diagnoses are currently only represented by OMIM terms, at the moment it is not possible to submit from this platform patients described by the Orphanet (https://www.orpha.net/consor/cgi-bin/index.php) or Decipher (https://www.deciphergenomics.org/) ontologies. As shown in Figure 1, gender might be optionally assigned to a patient to be submitted to the MME.

As regulations concerning genomic data sharing diverge depending on national legislation (Phillips, 2018), and even though initiatives like the EU's General Data Protection Regulation (GDPR) exist to harmonize the rules for data processing and sharing across borders in Europe and internationally (Bonomi et al., 2020), data controllers might not feel at ease disclosing the specific candidate variant(s) for a certain case (Molnár-Gabor and O Korbel, 2020). For this reason, we have included the option in Scout to describe MME patient's genotype features at the variant level (at the specific variant genotype level) or at the more generic gene level (only at the candidate gene level). These two options are illustrated by the last two lines in the patient's submission form of Figure 1.

The Scout user responsible for submitting a patient to MME network automatically becomes its contact person and will be notified if the submitted case is positively matched. The case data is stored in the database indefinitely and is subjected to internal and external queries, but can be reviewed (Figure 2), modified and eventually removed at any moment by the users.

MME nodes connected to PatientMatcher are displayed and can be searched independently for patients similar to the query case (external matching). Alternatively, similar patients can be also retrieved from the list of other Scout patients in PatientMatcher (internal matching) (Figure 3).

Both "active matches" (Scout patient has initiated the search and a matching patient has been found in a connected node or in PatientMatcher) and "passive matches" (an external party has initiated the matching and the Scout patient is among the results in PatientMatcher) are displayed in dedicated tabs named "Global Matches" and "Local matches", respectively displaying matching results for a patient against other nodes or other Scout patients (Figure 4).

## Software availability and installation

PatientMatcher is open-source and available on GitHub (https://github.com/Clinical-Genomics/patientMatcher). The software is distributed under the MIT license (https://github.com/Clinical-Genomics/patientMatcher/blob/master/LICENSE) and we encourage all interested parties to use and modify its code according to their needs. The main GitHub repository is curated by Clinical Genomics, but we look forward to establishing a collaborative environment where other users could help improving the code, adding or simply requesting new useful features.

The simplest way to run and test the server is to use the up-to-date container image with a basic software installation that can be pulled from Docker Hub (https://hub.docker.com/repository/docker/clinicalgenomics/patientmatcher). On the GitHub pages of the repository, we also provide instructions and support files to test PatientMatcher with real data without needing to install any software, except Docker. For this purpose, we compiled a multi-container Docker Compose file that, when launched by a single command from the terminal, provides a complete setup of the server, including a running instance of MongoDB containing the 50 benchmarking patients spanning 22 disorders described in Buske et al.(J. Buske et al., 2015). This server represents a standalone MME instance, ready to accept HTTP requests on localhost and port 9020. For development and testing reasons we have also created a more sophisticated Docker Compose setup, with an MME server connected to another two MME nodes (other instances of PatientMatcher), both containing demo patient data. This file is available under the /containers folder in the GitHub page of the PatientMatcher software. Deploying the software in a production environment using the official Docker image file could be achieved using Kubernetes (https://kubernetes.io/) or via Podman (https://podman.io/) system services. Another tested way to deploy the software is installing it from the Python Package Index (PyPI) using the Python installer Pip. In this case it is recommended to operate in a virtual environment, such as Conda (https://docs.conda.io/) after installing Python 3.6+. All these options, together with other server settings, are extensively described on the software GitHub pages.

## Conclusions

PatientMatcher is curated and maintained by Clinical Genomics, SciLifeLab in collaboration with the Genomic Medicine Center Karolinska at the Karolinska University Hospital, Stockholm. It is an open-source solution for clinical laboratories and research facilities who wish to join the federated MME network as independent nodes. Among the advantages of administering an independent node there is the control over the data submitted to the server. National legislation, for instance, might hinder storing sensitive data on cloud solutions or on servers located in other geographical areas. In a time of rapidly increasing genetic data generation, this MME implementation is meant to provide an easy-to administer tool to collect patient information and perform extensive comparisons between patients within the internal database or from external nodes. In order to reach as many users as possible we have designed a standalone application with customizable settings to harmonize the matching algorithm and notifications with data structures and routines present in different host centers. At the same time, we have established a pipeline where candidate variants or genes with linked patient phenotypes analyzed using the Scout decision support solution can be easily shared to the MME. In order to improve the code and better meet user expectations we look forward to collaborating with interested third parties to further develop the tool and its underlying matching algorithms. In conclusion, we look forward to teaming up with other clinical laboratories to share candidate gene-to-phenotype associations to contribute to the accelerating disease gene discovery.

**Acknowledgements**

We acknowledge the support to the BigMed project funded by the Norwegian Research Council and to the Genomic Medicine Sweden initiative funded by the Swedish Innovation Agency (Vinnova) both of which have financially contributed to the development of PatientMatcher.

# References

A. Philippakis A, R. Azzariti D, Beltran S, J. Brookes A, A. Brownstein C, Brudno M, G. Brunner H, J. Buske O, Carey K, Doll C, Dumitriu S, O.M. Dyke S, et al. 2015. The Matchmaker Exchange: A Platform for Rare Disease Gene Discovery. Human Mutation 36

Bonomi L, Huang Y, Ohno-Machado L. 2020. Privacy challenges and research opportunities for genomic data sharing. Nature Genetics 52

E. Soden S, J. Saunders C, K. Willig L, G. Farrow E, D. Smith L, E. Petrikin J, Baptiste LePichon J, A. Miller N, Thiffault I, L. Dinwiddie D, Twist G, Noll A, et al. 2014. Effectiveness of exome and genome sequencing guided by acuity of illness for diagnosis of neurodevelopmental disorders. Science Translational Medicine 6

Hamosh A, F. Scott A, Amberger J, Valle D, A. McKusick V. 2000. Online Mendelian Inheritance in Man (OMIM). Human Mutation 15

J. Buske O, Francois Schiettecatte, Hutton B, Dumitriu S, Misyura A, Huang L, Hartley T, Girdea M, Sobreira N, Mungall C, Brudno M. 2015. The Matchmaker Exchange API: Automating Patient Matching Through the Exchange of Structured Phenotypic and Genotypic Profiles. Human Mutation 36

Kohler S, C. Doelken S, J. Mungall C, Bauer S, V. Firth H, Bailleul-Forestier I, C.M. Black G, L. Brown D, Brudno M, Campbell J, R. Fitzpatrick D, T. Eppig J, et al. 2014. The Human Phenotype Ontology project: Linking molecular biology and disease through phenotype data. Nucleic Acids Research 42

L. Metzker M. 2010. Sequencing technologies the next generation. Nature Reviews Genetics 11

Lee H, L. Deignan J, Dorrani N, P. Strom S, Kantarci S, Quintero-Rivera F, Das K, Toy T, Harry B, Yourshaw M, Fox M, L. Fogel B, et al. 2014. Clinical exome sequencing for genetic identification of rare mendelian disorders. JAMA - Journal of the American Medical Association 312

M. Boycott K, R. Vanstone M, E. Bulman D, E. MacKenzie A. 2013. Rare-disease genetics in the era of next-generation sequencing: Discovery to translation. Nature Reviews Genetics 14

Molnar-Gabor F, O Korbel J. 2020. Genomic data sharing in Europe is stumbling—Could a code of conduct prevent its fall? EMBO Molecular Medicine 12

Pavan S, Rommel K, Maria Elena Mateo Marquina, Hohn S, Valerie Lanneau, Rath A. 2017. Clinical practice guidelines for rare diseases: The orphanet database. PLoS ONE 12

Pesquita C. 2007. Improving Semantic Similarity for Proteins based on the Gene Ontology. Genome.

Pesquita C, Faria D, Bastos H, Antonio E.N. Ferreira, Andre O. Falcao, M. Couto F. 2008. Metrics for GO based protein semantic similarity: A systematic evaluation. BMC Bioinformatics 9

Phillips M. 2018. International data-sharing norms: from the OECD to the General Data Protection Regulation (GDPR). Human Genetics 137

R. Azzariti D, Hamosh A. 2020. Genomic Data Sharing for Novel Mendelian Disease Gene Discovery: The Matchmaker Exchange. Annual Review of Genomics and Human Genetics 21

Shendure J, Ji H. 2008. Next-generation DNA sequencing : Article : Nature Biotechnology. Nature biotechnology 26

Stranneheim H, Lagerstedt-Robinson K, Mans Magnusson, Kvarnung M, Nilsson D, Lesko N, Engvall M, Marie Anderlid B, Arnell H, Backman Johansson C, Barbaro M, Bjorck E, et al. 2021. Integration of whole genome sequencing into a healthcare setting: high diagnostic rates across multiple clinical entities in 3219 rare disease patients. Genome Medicine 13

V. Firth H, M. Richards S, A. Paul Bevan, Clayton S, Corpas M, Rajan D, Van Vooren S, Moreau Y, M. Pettett R, P. Carter N. 2009. DECIPHER: Database of Chromosomal Imbalance and Phenotype in Humans Using Ensembl Resources. American Journal of Human Genetics 84

Yang Y, M. Muzny D, Xia F, Niu Z, Person R, Ding Y, Ward P, Braxton A, Wang M, Buhay C, Veeraraghavan N, Hawes A, et al. 2014. Molecular findings among patients referred for clinical whole-exome sequencing. JAMA - Journal of the American Medical Association 312

**Tables**

**Table 1.**

| Endpoint | Method | Rule | Purpose |
|---|---|---|---|
| add | POST | /patient/add | Adds or updates one patient by submitting a json payload s |
| delete | DELETE | patient/delete/<patient_id> | Deletes the patient with the given ID and all its matching re |
| heartbeat + | GET | /heartbeat | Returns a heartbeat response as defined in the MME API. |
| match_external | POST | /match/external/<patient_id> | Matches data from a patient already stored in PatientMatch |
| match_internal | POST | /match | Matches json data received from a request sent from a conne |
| matches | GET | /matches/<patient_id> | Returns all positive matches stored in the database for the p |
| metrics + | GET | /metrics | Returns a json object with server statistics described in the |
| nodes | GET | /nodes | Returns a response describing all connected nodes to the ser |

+ Endpoints not exposing sensitive information and therefore not requiring a security token to be accessed.

**Figure legends**

**Figure 1. MME patient submission form in Scout**

The submission of a Scout case to MatchMaker Exchange is initiated by clicking on a link present on the case page. The Scout user chooses which type of information (gender, HPO terms, OMIM terms, specific variant information or gene symbols only) will be submitted for the affected individuals of the case. Demo data was used to generate this figure.

**Figure 2. Overview of a MME patient in Scout**

A dedicated page in Scout summarizes the information associated to the patient submitted to MME. Demo data was used to generate this figure.

**Figure 3. Matching options selection**

MME nodes connected to PatientMatcher are displayed and can be searched independently for patients similar to the query case (external matching). Alternatively, similar patients can be also retrieved from the list of other Scout patients in PatientMatcher (internal matching). Demo data was used to generate this figure.

**Figure 4. Patient matching results page in Scout**

"Global Matches" and "Local Matches" tabs respectively display matching results for a patient against other nodes or other Scout patients. Red arrows designate the similarity score computed between query sample and matched sample. Demo data was used to generate this figure.