

1 Swarm Learning as a privacy-preserving machine learning 2 approach for disease classification

3
4 Stefanie Warnat-Herresthal^{1,*}, Hartmut Schultze^{2,*}, Krishnaprasad Lingadahalli Shastry^{2,*},
5 Sathyaranayanan Manamohan², Saikat Mukherjee², Vishesh Garg², Ravi Sarveswara²,
6 Kristian Händler^{3,*}, Peter Pickkers^{4,*}, N. Ahmad Aziz^{5,6,*}, Sofia Ktena^{7,*} Christian Siever²,
7 Michael Kraut³, Milind Desai², Bruno Monnet², Maria Saridaki⁷, Charles Martin Siegel², Anna
8 Drews³, Melanie Nuesch-Germano¹, Heidi Theis³, Mihai G. Netea^{8,9}, Fabian Theis¹⁰, Anna C.
9 Aschenbrenner^{1,8}, Thomas Ulas³, Monique M.B. Breteler^{5,11,#}, Evangelos J. Giannarellou-
10 Bourboulis^{7,#}, Matthijs Kox^{4,#}, Matthias Becker^{3,#}, Sorin Cheran^{2,#}, Michael S. Woodacre^{2,#}, Eng
11 Lim Goh^{2,#}, Joachim L. Schultze^{1,3,#}, German COVID-19 OMICS Initiative (DeCOI)

17 Affiliations:

- 18 1 Genomics and Immunoregulation, Life & Medical Sciences (LIMES) Institute, University
19 of Bonn, 53115 Bonn, Germany
20 2 Hewlett Packard Enterprise
21 3 PRECISE Platform for Single Cell Genomics and Epigenomics, German Center for
22 Neurodegenerative Diseases and the University of Bonn, 53127 Bonn, Germany
23 4 Department of Intensive Care Medicine and Radboud Center for Infectious Diseases
24 (RCI), Radboud University Medical Center, Nijmegen, 6500HB, The Netherlands
25 5 Population Health Sciences, German Center for Neurodegenerative Diseases (DZNE),
26 53175 Bonn, Germany
27 6 Department of Neurology, Faculty of Medicine, University of Bonn, 53127 Bonn, Germany
28 7 4th Department of Internal Medicine, National and Kapodistrian University of Athens,
29 Medical School, 124 62 Athens, Greece
30 8 Department of Internal Medicine and Radboud Center for Infectious Diseases (RCI),
31 Radboud University Medical Center, Nijmegen 6500HB, The Netherlands
32 9 Immunology & Metabolism, Life and Medical Sciences (LIMES) Institute, University of
33 Bonn, Bonn 53115, Germany
34 10 Institute of Computational Biology, Helmholtz Center Munich (HMGU), 85764
35 Neuherberg, Germany
36 11 Institute for Medical Biometry, Informatics and Epidemiology (IMBIE), Faculty of Medicine,
37 University of Bonn, 53175 Bonn, Germany

38
39
40
41 * shared first authors

42 # shared last authors

43 corresponding author j.schultze@uni-bonn.de

46 Key Words:

- 47 machine learning, artificial intelligence, Swarm Learning, decentralized machine learning,
48 privacy preserving machine learning, transcriptomics, blood, blood transcriptomics, acute
49 myeloid leukemia, acute lymphoblastic leukemia, COVID-19, tuberculosis

50 Abstract

51 Identification of patients with life-threatening diseases including leukemias or infections such
52 as tuberculosis and COVID-19 is an important goal of precision medicine. We recently
53 illustrated that leukemia patients are identified by machine learning (ML) based on their blood
54 transcriptomes. However, there is an increasing divide between what is technically possible
55 and what is allowed because of privacy legislation. To facilitate integration of any omics data
56 from any data owner world-wide without violating privacy laws, we here introduce Swarm
57 Learning (SL), a decentralized machine learning approach uniting edge computing,
58 blockchain-based peer-to-peer networking and coordination as well as privacy protection
59 without the need for a central coordinator thereby going beyond federated learning. Using
60 more than 14,000 blood transcriptomes derived from over 100 individual studies with non-
61 uniform distribution of cases and controls and significant study biases, we illustrate the
62 feasibility of SL to develop disease classifiers based on distributed data for COVID-19,
63 tuberculosis or leukemias that outperform those developed at individual sites. Still, SL
64 completely protects local privacy regulations by design. We propose this approach to
65 noticeably accelerate the introduction of precision medicine.

66

67

68 Introduction

69 Fast and reliable detection of patients with severe illnesses is a major goal of precision
70 medicine¹. The measurement of molecular phenotypes for example by omics technologies²
71 and the application of sophisticated bioinformatics including artificial intelligence (AI)
72 approaches^{3–7} opens up the possibility for physicians to utilize large-scale data for diagnostic
73 purposes in an unprecedented way. Yet, there is an increasing divide between what is
74 technically possible and what is allowed because of privacy legislation⁸ ([hhs.gov](https://www.hhs.gov),
75 <https://www.hhs.gov/hipaa/index.html>, 2020; Intersoft Consulting, General Data Protection
76 Regulation, <https://gdpr-info.eu>; Convention for the Protection of Individuals with regard to
77 Automatic Processing of Personal Data, <https://rm.coe.int/16808ade9d>). Particularly, in a
78 global health crisis, as in the case of the infection with severe acute respiratory syndrome
79 coronavirus 2 (SARS-CoV-2) leading to the pandemic spread of coronavirus disease 2019
80 (COVID-19)^{9–11}, reliable, fast, secure and privacy-preserving technical solutions based on AI
81 principles are now believed to add to the armamentarium to quickly answer important
82 questions in the fight against such threats^{12–15}. These AI-based concepts range from protein

83 structure prediction¹⁶, drug target prediction¹⁷, knowledge sharing¹⁸, tools for population
84 control^{19,20} to the assistance of healthcare personnel, e.g. by developing AI-based coronavirus
85 diagnostic software^{21,22}. Considering the more clinically oriented AI-based technical solutions,
86 any such progress might also induce improvements for a variety of deadly diseases including
87 other major infections or cancer²³. For example, the principles of a recently introduced AI-
88 system for diagnosing COVID-19 pneumonia and predicting disease outcome using computed
89 tomography²² might be further developed to identify patients with tuberculosis or lung cancer
90 in the future²⁴. At the same time, we need to consider important standards relating to data
91 privacy and protection, such as Convention 108(+) of the Council of Europe (Convention for
92 the Protection of Individuals with regard to Automatic Processing of Personal Data,
93 <https://rm.coe.int/16808ade9d>), which regulate the use and sharing of health data including in
94 AI-based approaches, irrespective of the occurrence of a pandemic crisis.

95 AI-based solutions intrinsically rely on appropriate algorithms²⁵, but even more so on large
96 enough datasets for training purposes²⁶. Since the domain of medicine is inherently
97 decentralized, the volume of data available locally is often insufficient to train reliable
98 classifiers^{27–29}. As a consequence, centralization of data, for example via cloud solutions, has
99 been one model to address the local limitations^{30–32}. While beneficial from an AI-perspective,
100 centralized solutions were shown to have other inherent hurdles, including increased data
101 traffic of large medical data, data ownership, privacy and security concerns when ownership
102 is disconnected from access and usage curation and thereby creating data monopolies
103 favoring data aggregators²⁶. Consequently, solutions to the challenges of central data models
104 in AI - particular when dealing with medical data - must be effective, with high accuracy and
105 efficiency, privacy- and ethics-preserving, secure, and fault-tolerant by design^{33–36}. Federated
106 AI has been introduced to address some of these aspects^{26,37–39}. While data are kept locally
107 (at the edge) and privacy issues are addressed^{40,41}, the model parameters in federated AI are
108 still handled by central custodians who as the intermediaries concentrate power of the learning
109 to themselves. Furthermore, such star-shaped architectures decrease fault tolerance.

110 We hypothesized that completely decentralized AI solutions overcome current technical
111 shortcomings and at the same time accommodate for inherently decentralized data structures
112 in medicine as well as pronounced data privacy and security regulations. The solution would
113 1) need to keep large medical data locally with the data owner, 2) require no raw data
114 exchange thereby also reducing data traffic and issues related to central storage, 3) provide
115 high level data security and privacy protection, 4) guarantee secure, transparent and fair
116 onboarding of decentralized members participating in the learning network without the need
117 for a central custodian, 5) allow for parameter merging with equal rights for all members
118 requiring no central custodian, and 6) protect the ML models from attacks. To address these

119 points, we introduce the concept of Swarm Learning (SL). SL combines decentralized
120 hardware infrastructures, distributed ML technique based on standardized AI engines with a
121 permissioned blockchain to securely onboard members, dynamically elect the leader among
122 the members, and merge model parameters. All processes are orchestrated by an SL library
123 and an iterative learning procedure applying AI solutions to compute problems with
124 decentralized private data.

125 Medicine is a prime example to illustrate the advantages of this AI approach. Without any
126 doubt, numerous medical features including radiograms or computed tomographies,
127 proteomes, metagenomes or microbiomes derived from body fluids including nasal or throat
128 swaps, blood, urine or stool are all excellently suitable medical data for the development of AI-
129 based diagnostic or outcome prediction classifiers. We here chose to evaluate the cellular
130 compartment of peripheral blood, either in form of peripheral blood mononuclear cells (PBMC)
131 or whole blood-derived transcriptomes, since blood-derived transcriptomes include important
132 information about the patients' immune response during a certain disease, which in itself is an
133 important molecular information^{42,43}. In other words, in addition to the use of blood-derived
134 high-dimensional molecular features for a diagnostic or outcome classification problem, blood
135 transcriptomes could be further utilized in the clinic to systematically characterize ongoing
136 pathophysiology, predict patient-specific drug targets and trigger additional studies targeting
137 defined cell types or molecular pathways, making this feature space even more attractive to
138 answer a wide variety of medical questions. Here, we illustrate that newly generated blood
139 transcriptome data together with data derived from more than 14,000 samples in more than
140 100 studies combined with AI-based algorithms in a Swarm Learning environment can be
141 successfully applied in real-world scenarios to detect patients with leukemias, tuberculosis or
142 active COVID-19 disease in an outbreak scenario across distributed datasets without the
143 necessity to negotiate and contractualize data sharing.

144

145

146 Results

147 Concept of Swarm Learning

148 Machine learning (ML) of any data including genome or transcriptome data requires the
149 availability of sufficiently large datasets^{26,44} and the respective compute infrastructure including
150 data storage for data processing and analytics⁴⁵. Conceptually, if data and compute
151 infrastructure is sufficiently available locally, ML can be performed locally ('at the edge') (**Fig.**
152 **1a**). However, often medical data are not sufficiently large enough locally and similar

153 approaches are performed at different locations in a disconnected fashion. These limitations
154 have been overcome by cloud computing where data are moved centrally to perform training
155 of ML algorithms in a centralized compute environment (**Fig. 1b**). Compared to local
156 approaches, cloud computing can significantly increase the amount of data for training ML
157 algorithms and therefore significantly improve their results²⁶. However, cloud computing has
158 other disadvantages such as data duplication from local to central data storage, increased
159 data traffic and issues with locally differing data privacy and security regulations⁴⁶. As an
160 alternative, federated cloud computing approaches such as Google's federated learning³⁸ and
161 Facebook's elastic averaging SGD (Deep learning with Elastic Averaging SGD,
162 <http://papers.neurips.cc/paper/5761-deep-learning-with-elastic-averaging-sgd.pdf>) have been
163 developed. In these models, dedicated parameter servers are responsible for aggregating and
164 distributing local learning (**Fig. 1c**). A disadvantage of such star-shaped system architectures
165 is the remainder of a central structure, which hampers implementation across different
166 jurisdictions and therefore still requires the respective legal negotiations. Furthermore, the risk
167 for a single point of failure at the central structure reduces fault-tolerance.

168 In an alternative model, which we introduce here as Swarm Learning (SL), we dismiss the
169 dedicated server and allow parameters and models to be shared only locally (**Fig. 1d**). While
170 parameters are shared via the swarm network, the models are built independently on private
171 data at the individual sites, here referred to as swarm edge nodes (short 'nodes') (**Fig. 1e**). SL
172 provides security measures to guarantee data sovereignty, security and privacy realized by a
173 private permissioned blockchain technology which enables different organizations or consortia
174 to efficiently collaborate (**Fig. 1f**). In a private permissioned blockchain network, each
175 participant is well defined and only pre-authorized participants can execute the transactions.
176 Hence, they use computationally inexpensive consensus algorithms, which offers better
177 performance and scalability. Onboarding of new members or nodes can be done dynamically
178 with the appropriate authorization measures to know the participants of the network, which
179 allows continuous scaling of learning (**Extended Data Fig. 1a**). A new node enrolls via a
180 blockchain smart contract, obtains the model, and performs local model training until defined
181 conditions for synchronization are met. Next, model parameters are exchanged via a Swarm
182 API with the rest of the swarm members and merged for an updated model with updated
183 parameter settings to start a new round of training at the nodes. This process is repeated until
184 stopping criterions are reached, which are negotiated between the swarm nodes/members.
185 The leader is dynamically elected using a blockchain smart contract for merging the
186 parameters and there is no need for a central coordinator in this swarm network. The
187 parameter merging algorithm is executed using a blockchain smart contract thus protects it
188 from semi-honest or dishonest participants. The parameters can be merged by the leader

189 using different functions including average, weighted average, minimum, maximum, or median
190 functions. The various merge techniques and merge frequency enables SL to efficiently work
191 with imbalanced and biased data. As currently developed, SL works with parametric models
192 with finite sets of parameters, such as linear regression or neural network models.

193 At each node, SL is conceptually divided into infrastructure and application layer (**Fig. 1g**). On
194 top of the physical infrastructure layer (hardware) the application environment contains the ML
195 platform, the blockchain, and the SL library (SLL) including the Swarm API in a containerized
196 deployment, which allows SL to be executed in heterogeneous hardware infrastructures (**Fig.**
197 **1g, Supplementary Information**). The application layer consists of the content, the models
198 from the respective domain, here medicine (**Fig. 1g**), for example blood transcriptome data
199 from patients with leukemias, tuberculosis and COVID-19 (**Fig. 1h-l**). Collectively, Swarm
200 Learning allows for a completely decentralized and therefore democratized, secure, privacy-
201 preserving, hardware-independent, and scalable machine learning environment, applicable to
202 many scenarios and domains, which we demonstrate with three medical examples.

203

204 **Swarm learning robustly predicts leukemias from peripheral blood mononuclear cell 205 data**

206 As a first use case, we chose transcriptomes derived from peripheral blood mononuclear cells
207 (PBMC) of more than 12,000 individuals (**Fig. 1h-j**) separated into three individual datasets
208 (A1, A2, A3) based on the technology used for generating the transcriptomes (2 different
209 microarrays, RNA-seq)⁴⁷. We used a deep neural network (Keras, <https://keras.io/>, 2015) as
210 the machine learning approach in all three use cases. To assess performance metrics of SL,
211 we simulated scenarios by dividing up the individual samples derived from several
212 independently performed studies (see Material and Methods) within each of the three datasets
213 into non-overlapping training and test sets. The training sets were then distributed to three
214 nodes for training and classifiers were tested at a fourth node (independent test set) (**Fig. 2a**).
215 By assigning the training data to the nodes in different distributions, we mimicked several
216 clinically relevant scenarios (**Supplementary Table 1**). As cases, we first used samples
217 defined as acute myeloid leukemia (AML), all other samples are termed 'controls'. Each node
218 within this simulation could stand for a large hospital or center, a network of hospitals
219 performing individual studies together, a country or any other independent institutional
220 organization generating such medical data with local privacy requirements.

221 In a first scenario, we randomly distributed samples per node as well as cases and controls
222 unevenly at the nodes and between nodes (dataset A2) (**Fig. 2b**). Sample distribution between
223 sample sets was permuted 100 times (**Fig. 2b, middle panel**) to determine the influence of

224 individual samples on overall performance. Among the nodes, the best test results were
225 obtained by node one with a mean accuracy of 97.0%, mean sensitivity of 97.5% and mean
226 specificity of 96.3% with an even distribution between cases and controls, albeit this node had
227 the smallest number of overall training samples. Node 2 did not produce any meaningful
228 results, which was due to a too low ratio of cases to controls (1:99) for training. Surprisingly,
229 node 3 with the largest number of samples, but an uneven distribution (70% cases : 30%
230 controls) performed worse than node 1 with a mean balanced accuracy of 95.1%. Most
231 importantly, however, SL outperformed each of the nodes resulting in a higher test accuracy
232 in 97.0% of all permutations (mean balanced accuracy 97.7%) (**Fig. 2b, right panel,**
233 **Supplementary Table 4**). The balanced accuracy of SL was significantly higher ($p < 0.001$)
234 when compared to the performance of each of the three nodes, despite the fact that
235 information from the poorly performing node 2 was integrated. We also calculated this scenario
236 in datasets A1 and A3 and obtained rather similar results strongly supporting that the
237 performance improvement of SL over single nodes is independent of data collection (studies)
238 and even experimental technologies (microarray (datasets A1, A2), RNA-seq (dataset A3)
239 used for data generation (**Extended Data Fig. 2**).

240 To test whether more evenly distributed samples at the nodes would improve individual node
241 performance, we distributed similar numbers of samples to each of the nodes but kept
242 case:control ratios as in scenario 1 (**Fig. 2c, Extended Data Fig. 3**). While there was a slight
243 increase in test accuracy at nodes 1 and 2, node 3 performed worse with also higher variance.
244 More importantly, SL still resulted in the best performance metrics (mean 98.5% accuracy)
245 with slightly but significantly ($p < 0.001$) increasing performance compared to the first scenario.
246 Results derived from datasets A1 and A3 echoed these findings (**Extended Data Fig. 3**).

247 In a third scenario, we distributed the same number of samples across all three nodes, but
248 increased potential batch effects between nodes, by distributing samples of a clinical study
249 independently performed and published in the past only to a dedicated training node. In this
250 scenario, cases and control ratios varied between nodes and left out samples (independent
251 samples) from the same published studies were combined for testing at node 4. Performance
252 of the three nodes was very comparable, but never reached SL results (mean 98.3% accuracy),
253 swarm outperformed all nodes with $p < 0.001$, **Fig. 2d., Extended Data Fig. 4b,**
254 **Supplementary Data Table 4**), which was also true for datasets A1 and A3 (**Extended Data**
255 **Fig. 4c-d**). Even when further increasing batch effects by distributing samples from
256 independent published studies to the test node, which means that training and test datasets
257 come from studies performed and published independently, SL outperformed the individual
258 nodes, albeit the variance in the results was increased both at each node and for SL, indicating

259 that study design has an overall impact on classifier performance and that this is still seen in
260 SL (mean 95.6% accuracy, **Extended Data Fig. 4e**).

261 In a fourth scenario, we further optimized the nodes by increasing the overall sample size at
262 node 3 and keeping case:control ratios even at all nodes (**Fig. 2e, Extended Data Fig. 5a-d**).
263 Clearly, node performance further improved with little variance between permutations,
264 however, even under these ‘node-optimized’ conditions, SL led to higher performance
265 parameters.

266 In a fifth scenario, we tested whether or not SL was ‘immune’ against the impact of the data
267 generation procedure (microarray versus RNA-seq) (**Fig. 2f, Extended Data Fig. 5e,f**). We
268 recently demonstrated that classifiers trained on data derived by one technology (e.g.
269 microarrays) do not necessarily perform well on another (e.g. RNA-seq)⁴⁷. To test this
270 influence on SL, we distributed the samples from the three different datasets (A1-A3) to one
271 node each, e.g. dataset A1 was used for training only at node 1. We used 20% of the data
272 (independent non-overlapping to the training data) from each dataset (A1-A3) and combined
273 them to form the test set (node 4). Node 3, trained on RNA-seq data, performed poorly on the
274 combined dataset due to the fact that two-thirds of the data in the test set were microarray-
275 derived data. Nodes 1 and 2 performed reasonably well with mean accuracies of 96.1% (node
276 1) and 97.5% (node 2), however did not reach the test accuracy of SL (98.8%), which also
277 indicated that SL is much more robust toward effects introduced by different data production
278 technologies in transcriptomics (**Fig. 2f, Extended Data Fig. 5e,f**).

279 Finally, we repeated several of these scenarios with acute lymphoblastic leukemia (ALL) as
280 the second most prevalent disease in dataset A2 (**Extended Data Fig. 6** and data not shown)
281 and demonstrated very similar results with SL outperforming the classifiers built at the nodes.
282 Collectively, these simulations using real-world transcriptome data collected from more than
283 100 individual studies illustrate that SL would not only allow data to be kept at the place of
284 generation and ownership, but it also outperforms every individual node in numerous
285 scenarios, even in those with nodes included that cannot provide any meaningful classifier
286 results.

287

288 **Swarm learning to identify patients with tuberculosis**

289 In infectious diseases, heterogeneity may be more pronounced compared to leukemia,
290 therefore we built a second use case predicting cases with tuberculosis (Tb) from full blood
291 transcriptomes. Of interest, previous work in smaller studies had already suggested that acute
292 tuberculosis or outcome of tuberculosis treatment can be revealed by blood transcriptomics

293 ^{48–52}. To apply SL, we generated a new dataset based on full blood transcriptomes derived by
294 PaxGene blood collection followed by bulk RNA-sequencing. We also generated new blood
295 transcriptomes and added existing studies to the dataset compiling a total of 1,999 samples
296 from nine individual studies including 775 acute and 277 latent Tb cases (**Fig. 1k, Extended**
297 **Data Fig. 7a, Supplementary Table 2**). These data are more challenging, since infectious
298 diseases show more variety due to biological differences with respect to disease severity,
299 phase of the disease or the host response. But also the technology itself is more variable with
300 numerous different approaches for full blood transcriptome sample processing, library
301 production and sequencing, which can introduce technical noise and batches between
302 studies. As a first scenario, we used all Tb samples (latent and acute) as cases and divided
303 Tb cases and controls evenly among the nodes (**Extended Data Fig. S7a-b, Supplementary**
304 **Table 1**). Similar to AML and ALL, in detecting Tb, SL outperformed the individual nodes in
305 accuracy (mean 93.4%), sensitivity (mean 96.0%) and specificity (mean 90.9%) (**Extended**
306 **Data Fig. S7b**). To increase the challenge, we decided to assess prediction of acute Tb cases
307 only. In this scenario, latent Tb are not treated as cases but rather as controls (**Extended Data**
308 **Fig. S7a**). For the first scenario, we kept cases and controls even at all nodes but further
309 reduced the number of training samples (**Fig. 3a-b**). As expected in this more challenging
310 scenario, distinguishing acute Tb from the control cohort (including latent Tb samples), overall
311 performance (mean balanced accuracy 89.1%, mean sensitivity 92.2%, mean specificity
312 86.0%) slightly dropped, but still SL performed better than any of the individual nodes ($p < 0.01$
313 for swarm vs. each node, **Fig. 3b**). To determine whether sample size impacts on prediction
314 results in this scenario, we reduced the number of samples at each training node (1-3) by
315 50%, but kept the ratio between cases and controls (**Extended Data Fig. S7c**). Still, SL
316 outperformed the nodes, but all statistical readouts (mean accuracy 86.5%, mean sensitivity
317 87.8%, mean specificity 84.8%) at all nodes and SL showed lower performance, following
318 general observations of AI with better performance when increasing training data²⁶. We next
319 altered the scenario by dividing up the three nodes into six smaller nodes (**Fig. 3c**, samples
320 per node reduced by half in comparison to **Fig. 3a-b**), a scenario that can be envisioned in the
321 domain of medicine in many settings, for example if several smaller medical centers with less
322 cases would join efforts (**Fig. 3d**). Clearly, each individual node performed worse, but for SL
323 the results did not deteriorate (mean accuracy 89.2%, mean sensitivity 90.7%, mean
324 specificity 88.2% with significant difference to each of the nodes in all performance measures,
325 see **Supplementary Table 4**), again illustrating the strength of the joined learning effort, while
326 completely respecting each individual node's data privacy.

327 Albeit aware of the fact that - in general - acute Tb is an endemic disease and does not tend
328 to develop towards a pandemic such as the current COVID-19 pandemics, we utilized the Tb

329 blood transcriptomics dataset to simulate potential outbreak and epidemic scenarios to
330 determine benefits, but also potential limitations of SL and how to address them (**Fig. 3e-l**).
331 The first scenario reflects a situation in which three independent regions (simulated by the
332 nodes), would already have sufficient but different numbers of disease cases. Furthermore,
333 cases and controls were kept even at the test node (**Fig. 3e-f**). Overall, compared to the
334 scenario described in **Fig. 3c**, results for the swarm were almost comparable (mean accuracy
335 89.0%, mean sensitivity 94.4%, mean specificity 83.4%), while the results for the node with
336 the lowest number of cases and controls (node 2) dropped noticeable (mean accuracy 82.2%,
337 mean sensitivity 88.8%, mean specificity 75.4%, **Fig. 3f**). When reducing the prevalence at
338 the test node by increasing the number of controls (**Fig. 3g-h**), this effect was even more
339 pronounced, while the performance of the swarm was almost unaffected (mean balanced
340 accuracy 89.0%).

341 We decreased the number of cases at a second training node (node 1) (**Fig. 3i-l**), which clearly
342 reduced test performance for this particular node (**Fig. 3i-j**), while test performance of the
343 swarm was only slightly inferior to the prior scenario (mean balanced accuracy 87.5%, no
344 significant difference to the prior scenario). Only when reducing the prevalence at the test
345 node (**Fig. 3k-l**), we saw a further drop in mean specificity for the swarm (81.0%), while
346 sensitivity stayed similarly high (93.0%). Finally, we further reduced the prevalence at two
347 training nodes (node 2: 1:10; node 3: 1:5) as well as the test node (**Extended Data Fig. 8a-**
348 **b**). Lowering the prevalence during training resulted in very poor test performance at these
349 two nodes (balanced accuracy node 2: 59.8%, balanced accuracy node 3: 74.8%), while
350 specificity was high (node 2: 98.4%, node 3: 93.8%). SL showed highest accuracy (mean
351 balanced accuracy 86.26%) and F-statistics (90.0%) but was outperformed for sensitivity by
352 node 1 (swarm: 80.0%, node1: 87.8%), which showed poor performance concerning
353 specificity (swarm: 92.4%, node1: 84.8%). Vice versa, node 2 outperformed the swarm for
354 specificity (98.4%), but showed very poor sensitivity (21.2%) (**Extended Data Fig. 8b**). When
355 lowering prevalence at the test node (**Extended Data Fig. 8c-d**), it became clear that all
356 performance parameters including the F1 statistics were more resistant for the swarm
357 compared to individual nodes. Taken together, using whole blood transcriptomes instead of
358 PBMC and acute Tb as the disease instead of leukemia, we present a second use case
359 illustrating that Swarm Learning integrating several individual nodes outperforms each node.
360 Furthermore, we gained initial insights into the potential of SL to be utilized in a disease
361 outbreak scenario.

362

363 **Identification of COVID-19 patients in an outbreak scenario**

364 Based on the promising results obtained for tuberculosis, we collected blood from COVID-19
365 patients at two sites in Europe (Athens, Greece; n=39 samples, Nijmegen, n=93 samples) and
366 generated whole blood transcriptomes by RNA-sequencing. We used the dataset described
367 for Tb as the framework and included the COVID-19 samples (**Fig. 1I**) for assessing whether
368 SL could be applied early on to detect patients with a newly identified disease. While COVID-
369 19 patients are currently identified by PCR-based assays to detect viral RNA⁵³, we use this
370 case as a proof-of-principle study to illustrate how SL could be used even very early on during
371 an outbreak based on the patients' immune response captured by analysis of the circulating
372 immune cells in the blood. Here, blood transcriptomes only present a potential feature space
373 to illustrate the performance of SL. Furthermore, assessing the specific host response, in
374 addition to disease prediction, might be beneficial in situations for which the pathogen is
375 unknown, specific pathogen tests not yet possible, and blood transcriptomics can contribute
376 to the understanding of the host's immune response⁵⁴. Lastly, while we do not have the power
377 yet, blood transcriptome-based machine learning might be used to predict severe COVID-19
378 cases, which cannot be done by viral testing alone.

379 COVID-19 induces very strong changes in peripheral blood transcriptomes⁵⁴. Following our
380 experience with the leukemia and tuberculosis use cases, we first tested classifier
381 performance for evenly distributed cases and controls at both training nodes and the test node
382 (**Extended Data Fig. 9a,b, Supplementary Table 1**). We reached very high statistical
383 performance parameters, including high F1-statistics with SL showing highest mean values
384 for accuracy (96.4%), sensitivity (97.8%), and F1 score (96.4%) (**Extended Data Fig. 9b**,
385 summary statistics for all figures are listed in **Supplementary Table 4**). Reducing the
386 prevalence at the test node (11:25 cases:controls) reduced all test parameters (**Extended**
387 **Data Fig. 9c**), but only when we reduced the prevalence even further (1:44 ratio, **Extended**
388 **Data Fig. 9d**), F1-statistics was clearly reduced, albeit SL again performing best. We next
389 reduced the cases at all training nodes (**Extended Figure 10**), but even under these
390 conditions, we observed still very high values for accuracy, sensitivity, specificity and F1
391 scores, both derived by training at individual nodes or by SL (**Extended Figure 10a-f**).
392 We then reduced the cases at all three training nodes to very low numbers, a scenario that
393 might be envisioned very early during an outbreak scenario (**Fig. 4a**). Node 1 contained only
394 20 cases, node 2 10 cases and node 3 only 5 cases. At each node, controls outnumbered
395 cases by 1:5, 1:10, or 1:20. At the test node, we varied the prevalence from 1:1 (**Fig. 4b**), 1:2
396 (**Fig. 4c**) to 1:10 (**Fig. 4d**). Based on our findings for Tb (**Extended Data Fig. 8**), we expected
397 classifier performance to deteriorate under these conditions. We only observed decreased
398 performance at nodes 2 and 3 in these scenarios with SL outperforming these nodes with
399 p<0.05 for all performance measures, e.g. at a test node prevalence of 1:10 (accuracy

400 (99.3%), sensitivity (95.1%), specificity (99.7%) and F1-statistics (99.7%) (**Fig. 4d**). Finally,
401 we simulated a scenario with four instead of three training nodes with very few cases per node
402 (**Extended Data Fig. 11a-d**), in an otherwise similar scenario as described for Fig. 4. Even for
403 a simulated prevalence of 1:10 cases versus controls at the test node, we determined high
404 test performance parameters for SL, with swam performance being significantly higher than
405 node performances (SL accuracy (99.1%), sensitivity (92.0%), specificity (99.9%), F1 statistics
406 (99.7%) (**Extended Data Fig. 11**) with the lowest variance in performance, while results at
407 individual notes were very variable and deteriorated with low case numbers at the training
408 node. Collectively, we provide first evidence that blood transcriptomes taken from patients with
409 COVID-19 harbor very strong biological changes and these translate into a very powerful
410 feature space for applying machine learning to the detection of patients with this new infectious
411 disease, particularly when applying SL.

412

413 Discussion

414 The introduction of precision medicine based on high-resolution molecular and imaging data
415 will heavily rely on trustworthy machine learning algorithms in compute environments that are
416 characterized by high accuracy and efficiency, that are privacy- and ethics-preserving, secure,
417 and that are fault-tolerant by design³³⁻³⁶. At the same time, privacy legislation is becoming
418 increasingly strict, as risks of cloud-based and central data-acquisition are recognized. Here,
419 we introduce Swarm Learning, which combines blockchain technology and machine learning
420 environments organized in a swarm network architecture with independent swarm edge nodes
421 that harbor local data, compute infrastructure and execute the shared learning models that
422 make central data acquisition obsolete. During iterations of SL, one of the nodes is chosen to
423 lead the iteration, which does not require a central parameter server anymore thereby
424 restricting centralization of learned knowledge and at the same time increasing resiliency and
425 fault tolerance. In fact, these are the most important improvements over current federated
426 computing models. Furthermore, private permissioned blockchain technology, harboring all
427 rules of interaction between the nodes, is the Swarm Learning's inherent privacy- and ethics-
428 preserving strategy. This is of particular interest to medical data and could be adapted by other
429 federated learning systems. To understand whether the concept of swarm learning would also
430 be characterized by high efficiency and high accuracy, we built three medical use cases based
431 on blood transcriptome data, which are high-dimensional data derived from blood, one of the
432 major tissues used for diagnostic purposes in medicine. First, utilizing three previously
433 compiled datasets (A1-3) of peripheral blood mononuclear cells derived from patients with
434 acute myeloid leukemia, we provide strong evidence that SL-based classifier generation using

435 a well-established neural network algorithm outperforms individual nodes, even in scenarios
436 where individual contributing swarm nodes were performing rather poorly. Most striking,
437 swarm learning was even improving performance parameters when training of individual
438 nodes was based on technically different data, a situation that was previously shown to
439 deteriorate classifier performance⁴⁷. With these promising results, we generated a more
440 challenging use case in infectious disease patients, detecting Tb based on full blood
441 transcriptomes. Also in this case, SL outperformed individual nodes. Using Tb to simulate
442 scenarios that could be envisioned for building blood transcriptome classifiers for patients
443 during an outbreak situation, we further illustrate the power of SL over individual nodes.
444 Considering the difficulty to quickly negotiate data sharing protocols or contracts during an
445 epidemic or pandemic outbreak, we deduce from these findings that SL would be an ideal
446 strategy for independent producer of medical data to quickly team up to increase the power to
447 generate robust and reliable machine learning-based disease or outcome prediction classifier
448 without the need to share data or relocate data to central cloud storages.

449 In addition, we tested whether we could build a disease prediction classifier for COVID-19 in
450 an outbreak scenario. Building on our knowledge that blood transcriptomes of COVID-19
451 patients are significantly altered with hundreds of genes being changed in expression and with
452 a rather specific signature compared to other infectious diseases⁵⁴, we hypothesized that it
453 should be possible to build such a classifier with a rather small number of samples. Here, we
454 provide evidence that classifiers with high accuracy, sensitivity, specificity, and also high F1-
455 statistics can be generated to identify patients with COVID-19 based on their blood
456 transcriptomes. Moreover, we illustrate the power of SL that would allow to quickly increase
457 the power of classifier generation even under very early outbreak scenarios with very few
458 cases used at the training nodes, which could be e.g. collaborating hospitals in an outbreak
459 region. Since data do not have to be shared, additional hospitals could benefit from such a
460 system by applying the classifiers to their new patients and once classified, one could even
461 envision an onboarding of these hospitals for an adaptive classifier improvement schema.
462 Albeit technically feasible, we are fully aware that such scenarios require further classifier
463 testing and confirmation, but also an assessment of how this could be integrated in existing
464 legal and ethical regulations at different regions in the world^{5,6}. Furthermore, we appreciate
465 that other currently less expensive data might be suitable for generating classifiers to identify
466 COVID-19 patients¹⁰. For example, if highly standardized clinical data would become
467 available, SL could be used to interrogate the clinical feature space at many clinics worldwide
468 without any need to exchange the data to develop high performance classifiers for detecting
469 COVID-19 patients. Similarly, recently introduced AI-systems using imaging data^{21,22} might be
470 more easily scaled if many hospitals with such data could be connected via SL. Irrespective

471 of these additional opportunities using other parameter spaces, we would like to suggest blood
472 transcriptomics as a promising new alternative due to its very strong signal in COVID-19. A
473 next step will be to determine whether blood transcriptomes taken at early time points could
474 be used to predict severe disease courses, which might allow physicians to introduce novel
475 treatments at an earlier time point. Furthermore, we propose to develop an international
476 database of blood transcriptomes that could be utilized for the development of predictive
477 classifiers in other infectious and non-infectious diseases as well. It could be envisioned that
478 such an SL-based learning scheme could be deployed as a permanent monitoring or early
479 warning system that runs by default, looking for unusual movements in molecular profiles.
480 Collectively, SL together with transcriptomics but also other medical data is a very promising
481 approach to democratize the use of AI among the many stakeholders in the domain of
482 medicine while at the same time resulting in more data privacy, data protection and less data
483 traffic.

484 With increasing efforts to enforce data privacy and security of medical data⁸ ([hhs.gov](https://www.hhs.gov),
485 <https://www.hhs.gov/hipaa/index.html>, 2020; Intersoft Consulting, General Data Protection
486 Regulation, <https://gdpr-info.eu>) and to reduce data traffic and duplication of large medical
487 data, a decentralized data model will become the preferred choice of handling, storing,
488 managing and analyzing medical data²⁶. This will not be restricted to omics data as exemplified
489 here, but will extend to other large medical data such as medical imaging data^{55,56}. Particularly
490 in oncology, great successes applying machine learning have already been reported for tumor
491 detection^{47,55,57,58}, subtyping^{59,60}, grading⁶¹, genomic characterization⁶², or outcome
492 prediction⁶³, yet progress is hindered by too small datasets at any given institution²⁶ with
493 current privacy regulations⁸ ([hhs.gov](https://www.hhs.gov), <https://www.hhs.gov/hipaa/index.html>, 2020; Intersoft
494 Consulting, General Data Protection Regulation, <https://gdpr-info.eu>) making it less appealing
495 to develop centralized AI systems. We introduce Swarm Learning as a decentralized learning
496 system with access to data stored locally that can replace the current paradigm of data sharing
497 and centralized storage while preserving data privacy in cross-institutional research in a wide
498 spectrum of biomedical disciplines. Furthermore, SL can easily inherit developments to further
499 preserve privacy such as functional encryption⁶⁴, or encrypted transfer learning approaches⁶⁵.
500 In addition, the blockchain technology applied here provides robust measures against semi-
501 honest or dishonest participants/adversaries who might attempt to undermine a Swarm
502 Network. Another important aspect for wide employment of SL in the research community and
503 in real-world applications is the ease of use of the Swarm API, which will make it easier for
504 researchers and developers to include novel developments such as for example private
505 machine learning in TensorFlow⁶⁶.

506 There is no doubt that numerous medical and other data types as well as a vast variety of
507 computational approaches can be used during a pandemic¹⁴. We do not want to imply that
508 blood transcriptomics would be the preferred solution for the many questions that AI and
509 machine learning could help to solve during such a crisis. Although, at the same time, we have
510 recently shown that blood transcriptomics can be used to define molecular phenotypes of
511 COVID-19, uncover the deviated immune response in severe COVID-19 patients, define
512 unique patterns of the disease in comparison to other diseases and can be utilized to predict
513 potential drugs to be repurposed for COVID-19 therapy (Aschenbrenner et al. unpublished
514 results). Therefore, we explored blood transcriptomics as a unique and rich feature space and
515 a good example to illustrate the advantages of SL in identifying COVID-19 patients. Once
516 larger datasets become available, SL could be used to identify patients at risk to develop
517 severe COVID-19 early after onset of symptoms.

518 Another important quest that has been proposed is global collaboration and data-sharing¹³.
519 While we could not agree more about the need for global collaboration - an inherent
520 characteristic of SL - we favor systems that do not require data sharing but rather support
521 global collaboration with complete data privacy preservation. Particularly, if using medical data
522 that can also be used to interrogate medical issues unrelated to COVID-19. Indeed,
523 statements by lawmakers have been triggered, clearly indicating that privacy rules also fully
524 apply during the pandemics (EU Digital Solidarity: a call for a pan-European approach against
525 the pandemic, Wojciech Wiewiórowski, https://edps.europa.eu/sites/edp/files/publication/2020-04-06_eu_digital_solidarity_covid19_en.pdf, 2020). Particular in a crisis situation such
527 as the current pandemic, AI systems need to comply with ethical principles and respect human
528 rights¹⁴. We therefore argue that systems such as Swarm Learning that allow fair, transparent
529 and still highly regulated shared data analytics while preserving data privacy regulations are
530 to be favored, particularly during times of high urgency to develop supportive tools for medical
531 decision making. We therefore also propose to explore SL for image-based diagnostics of
532 COVID-19 from patterns in X-ray images or computed tomography (CT) scans^{21,22}, structured
533 health records⁶⁷, or wearables for disease tracking¹⁴. Swarm learning would also have the
534 advantage that model and code sharing as well as dissemination of new applications is easily
535 scalable, because onboarding of new swarm participants is structured by blockchain
536 technology, while scaling of data sharing is not even necessary due the inherent local
537 computing of the data¹⁴. Furthermore, swarm learning can reduce the burden of establishing
538 global, comprehensive, open, and verified datasets.

539 Collectively, we introduce Swarm Learning defined by the combination of blockchain
540 technology and decentralized machine learning in an entirely democratized approach

541 eliminating a central player and therefore representing a uniquely fitting strategy for the
542 inherently locally organized domain of medicine. We used blood transcriptomes in three
543 scenarios as use cases since they combine blood as the most widely used surrogate tissue
544 for diagnostic purposes with an omics technology producing high-dimensional data with many
545 parameters. Since the deployment of Swarm Learning due to ease of use of Swarm Learning
546 libraries is a rather simple task, we propose to expand the use of this technology and further
547 develop such classifiers in a unifying fashion across centers worldwide without any need to
548 share the data itself. Our use cases are supposed to serve as examples for other high-
549 dimensional data in the domain of medicine, but certainly also many other areas of research
550 and application against the pandemics and beyond.

551 **Acknowledgments:**

552 We thank Dr. Sach Mukherjee (Statistics and Machine Learning, German Center for
553 Neurodegenerative Diseases) for feedback on the manuscript.

554

555 **Deutsche COVID-19 Omics Initiative (DeCOI)**

556 Robert Bals, Alexander Bartholomäus, Anke Becker, Ezio Bonifacio, Peer Bork, Thomas
557 Clavel, Maria Colome-Tatche, Andreas Diefenbach, Alexander Dilthey, Nicole Fischer, Konrad
558 Förstner, Julien Gagneur, Alexander Goesmann, Torsten Hain, Michael Hummel, Stefan
559 Janssen, René Kallies, Birte Kehr, Andreas Keller, Sarah Kim-Hellmuth, Christoph Klein,
560 Oliver Kohlbacher, Jan Korbel, Ingo Kurth, Markus Landthaler, Yang Li, Kerstin Ludwig, Oliwia
561 Makarewicz, Manja Marz, Alice McHardy, Christian Mertes, Markus Nöthen, Peter Nürnberg,
562 Uwe Ohler, Stephan Ossowski, Jörg Overmann, Klaus Pfeffer, Alfred Pühler, Nikolaus
563 Rajewsky, Markus Ralser, Olaf Rieß, Stephan Ripke, Ulisses Nunes da Rocha, Philip
564 Rosenstiel, Antoine-Emmanuel Saliba, Leif Erik Sander, Birgit Sawitzki, Philipp Schiffer,
565 Joachim L. Schultze, Alexander Sczyrba, Oliver Stegle, Jens Stoye, Fabian Theis, Janne
566 Vehreschild, Jörg Vogel, Max von Kleist, Andreas Walker, Jörn Walter, Dagmar Wieczorek,
567 John Ziebuhr

568

569

570 **Funding:**

571 This work was supported in part by the German Research Foundation (DFG) to J.L.S. (INST
572 37/1049-1, INST 216/981-1, INST 257/605-1, INST 269/768-1 and INST 217/988-1, INST
573 217/577-1); the HGF grant sparse2big, the EU projects SYSCID (grant number 733100) and
574 ImmunoSep, and by HPE to the DZNE for generating whole blood transcriptome data from
575 patients with COVID-19. Sofia Ktena is supported by the Hellenic Institute for the Study of
576 Sepsis. The clinical study in Greece was supported by the Hellenic Institute for the Study of
577 Sepsis. E.J.G.-B. received funding from the FrameWork 7 program HemoSpec (granted to the
578 National and Kapodistrian University of Athens), the Horizon2020 Marie-Curie Project
579 European Sepsis Academy (granted to the National and Kapodistrian University of Athens),
580 and the Horizon 2020 European Grant ImmunoSep (granted to the Hellenic Institute for the
581 Study of Sepsis).

582 Author contributions:

583 Conceptualization, J.L.S; K.L.S, S.Ma.; Methodology, S.W.-H., C.S., R.S., M.D., B.M., C.M.S.;
584 Software: M.L.; V.G, C.S., S.Ma., S.Mu.; Investigation, S.W.-H., K.S., M.K., A.D., M.N., H.T.,
585 T.U.; Biospecimen/ enzyme resources, M.K., P.P., M.G.N., S.K., E.G.-B., M.M.B.BM., C.K.,
586 M.G.; Writing – Original Draft, S.W.-H., H.S., K.L.S., M.B., J.L.S.; Writing – Review & Editing,
587 S.W.-H., H.S., K.L.S., D.S., A.C.A., M.K., P.P., M.G.N., M.B., S.C., M.S.W., E.L.G., J.L.S.;
588 Visualization, H.S., J.L.S; Supervision, H.S., K.L.S., J.L.S.; Project Administration, H.S.,
589 J.L.S.; Funding Acquisition, W.F., J.L.S., F.J.T., J.H., N.Y., A.K.S., M.G., and M.M.B.B.

590

591

592 Ethics declarations

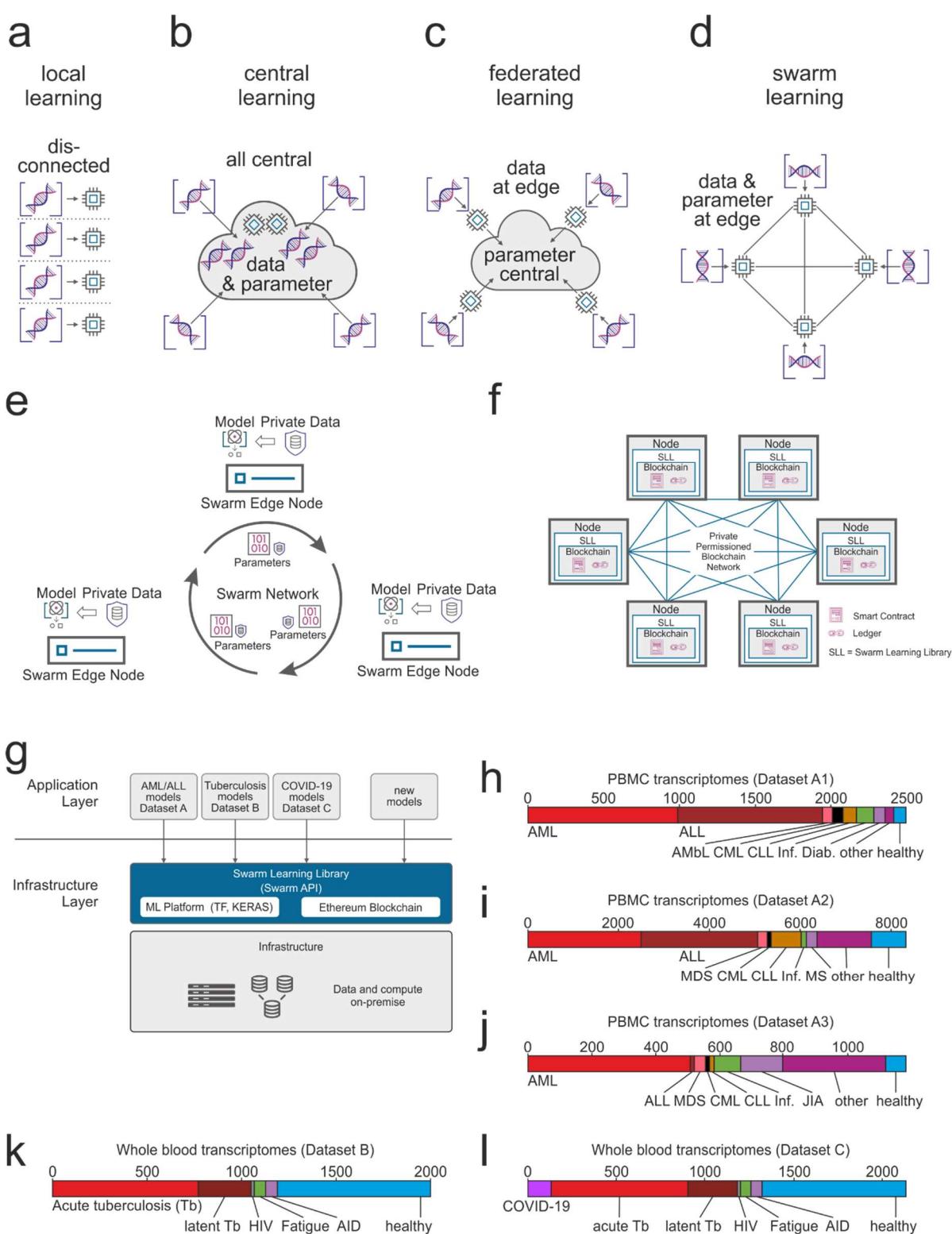
593 Competing Interests

594 H.S., K.L.S, S.Ma., S.Mu., V.G., R.S., C.S., M.D., B.M, C.M.S., S.C., M.S.W, E.L.G are
595 employees of Hewlett Packard Enterprise. Hewlett Packard Enterprise developed the Swarm
596 Learning Library in its entirety as described in this work and has submitted multiple associated
597 patent applications. E.J.G.-B. received honoraria from AbbVie USA, Abbott CH, InflaRx
598 GmbH, MSD Greece, XBiotech Inc. and Angelini Italy; independent educational grants from
599 AbbVie, Abbott, Astellas Pharma Europe, AxisShield, bioMérieux Inc, InflaRx GmbH, and
600 XBiotech Inc.

601 Figures and Figure Legends

602

Figure 1



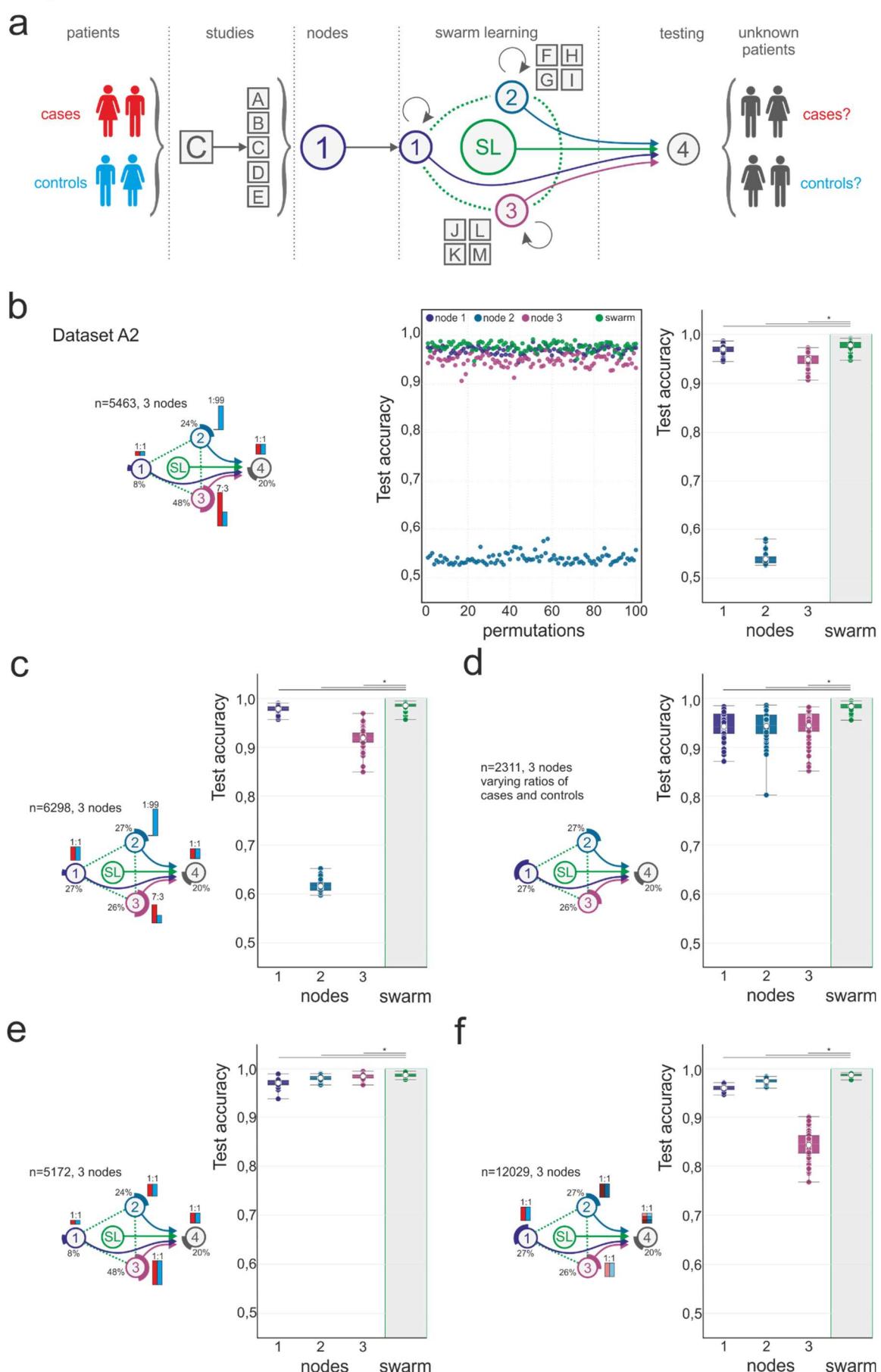
603

604

605 **Figure 1. Concept of Swarm Learning**

606 **(a-d)** The principles of Swarm Learning in contrast to other machine learning concepts. **(a)**
607 Illustration of the concept of local learning with data and computation at different, but
608 disconnected locations. **(b)** Principle of cloud-based machine learning where data from
609 contributing centers move copies of the data to a central cloud-based storage; centrally
610 located data are then used for central - often cloud-based - machine learning. **(c)** Federated
611 learning with data being kept with the data contributor and computing is also performed at the
612 site of local data storage and availability, yet parameter settings of machine learning are
613 orchestrated by a central parameter server. **(d)** Swarm Learning principle with swarm nodes
614 being connected in a democratic fashion (enabled by blockchain technology) without the need
615 for a central custodian or parameter server. Data privacy is preserved, data is kept where it is
616 generated, computation is achieved locally and learning parameters are shared within the
617 Swarm Network. **(e)** Schematic representation of the Swarm Network consisting of the Swarm
618 Edge Nodes (short 'nodes') that exchange parameters for learning, which is implemented
619 using blockchain technology. Use of private data at each node together with the model
620 provided via Swarm Network. **(f)** Concept and outline of the private permissioned blockchain
621 network as a layer of the Swarm Learning network. Each node consists of the blockchain,
622 including the ledger and smart contract, as well as the Swarm Learning Library (SLL) with the
623 API to interact with other nodes within the network. **(g)** Application and infrastructure layer as
624 part of the Swarm Learning concept. **(h-l)** Description of the transcriptome datasets used
625 within this study: Dataset **(h)** A1 and **(i)** A2, two microarray-based transcriptome datasets of
626 peripheral blood mononuclear cells (PBMC). **(j)** Dataset A3, RNA-seq based transcriptomes
627 of PBMC. Dataset **(k)** B and **(l)** C, RNA-seq based whole blood transcriptome datasets.
628 Abbreviations: *AML*, Acute Myeloid Leukemia; *ALL*, Acute Lymphoblastic Leukemia; *COVID-19*,
629 CoronaVirus Disease 2019; *API*, Application Programming Interface; *ML*, Machine
630 Learning; *TF*, Tensor Flow; *KERAS*, Open Source Deep Learning Library; *AMbl*, Acute
631 Myeloblastic Leukemia; *CML*, Chronic Myeloid Leukemia; *CLL*, Chronic Lymphocytic
632 Leukemia; *Inf.*, Infections, *Diab.*, Diabetes Type II; *MDS*, Myelodysplastic Syndrome; *MS*,
633 multiple sclerosis; *JIA*, Juvenile idiopathic arthritis; *Tb*, tuberculosis; *HIV*, Human
634 Immunodeficiency Virus, *AID*, Acute Infectious Disease. SLL Swarm Learning Library.

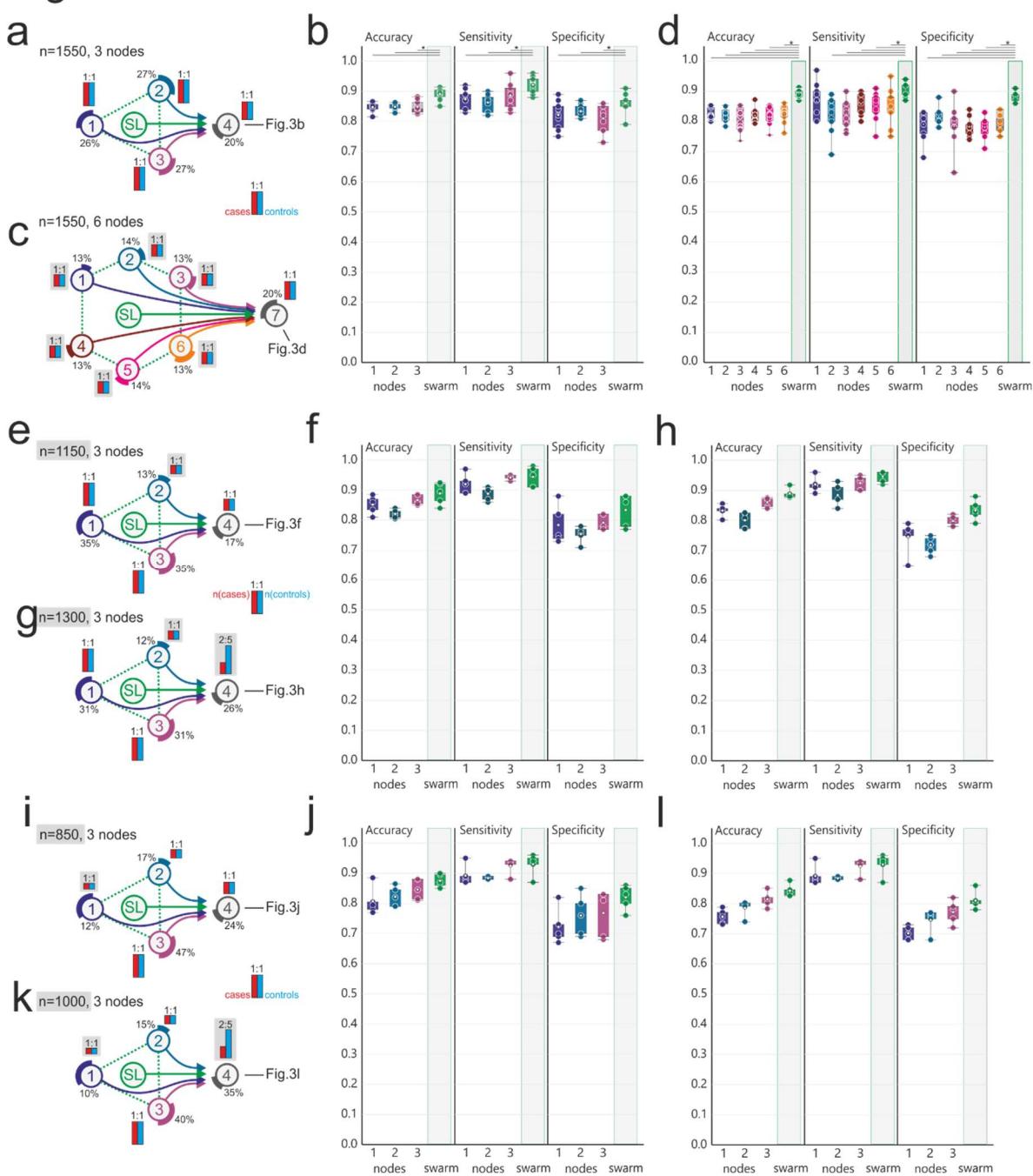
Figure 2



636 **Figure 2. Swarm learning to predict leukemias from PBMC data**

637 **(a)** Schematic representation of the use of the transcriptome data derived from more than
638 12,000 individuals in over 100 individual studies⁴⁷. Principle of distribution of data to individual
639 Swarm Edge Nodes (short ‘nodes’). Nodes 1-3 were used for training, node 4 for testing.
640 Swarm Learning (SL) was achieved by integrating nodes 1-3 for training following procedures
641 described in detail in Supplementary Information. **(b)** Scenario using Dataset A2. Left panel
642 illustrating the setting of the scenario concerning distribution of cases and controls to individual
643 nodes, as well as total number of samples used for this scenario. Cases (red bar) and controls
644 (blue bar) were distributed unevenly among nodes, the number of samples distributed to each
645 node was also uneven in this scenario. Middle panel shows results of accuracy of all 100
646 permutations performed for the 3 training nodes individually as well as the results obtained by
647 SL. Accuracy is defined for the independent fourth node used for testing only. Right panel
648 represents box-whisker plot representation of the individual data presented in the middle panel
649 showing mean, 1st and 3rd quartile and whisker type Min/Max. **(c)** Scenario with uneven
650 numbers of cases and controls at the different training nodes but similar numbers of samples
651 at each node to determine impact of these changes on SL performance. Left panel: schematic
652 representation of scenario and right panel: results obtained for accuracy at the test node (node
653 4) for each of the three training nodes 1-3 and SL independently as box and whisker plot with
654 the same parameter as described for (b). **(d)** Scenario with even numbers at each of the
655 nodes, schematic representation (left panel) and visualization of results as box-whisker plots
656 as in (b) and (c). **(e)** Scenario with even distribution of cases and controls at each training
657 node, but different numbers of samples at each node and overall increase in numbers of
658 samples. Representation of schema and data visualization as in (b-d). **(f)** Scenario where each
659 node obtained samples from different Datasets (node 1: Dataset A1, node 2: Dataset A2, node
660 3, Dataset A3). Node 4 obtained samples from each Dataset A1-A3 to define impact on
661 technical bias on Swarm Learning performance. Representation of schema and data
662 visualization as in (b-e). Statistical differences between results derived by SL and individual
663 nodes including all permutations performed were calculated with Wilcoxon signed rank test
664 with continuity correction; asterisk and line: p<0.05.

Figure 3



665

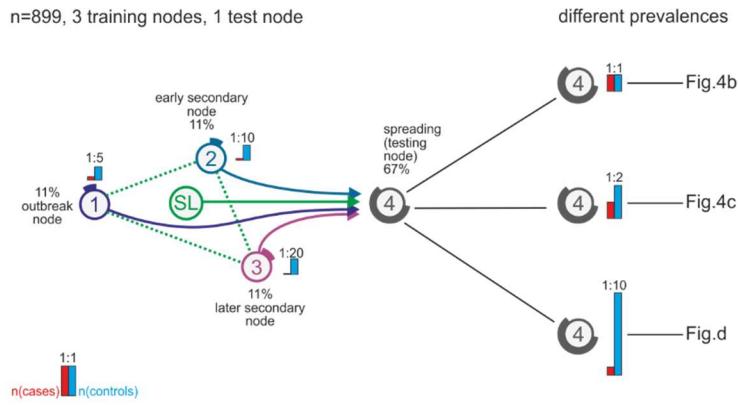
666

667 **Figure 3. Swarm learning to identify patients with tuberculosis**

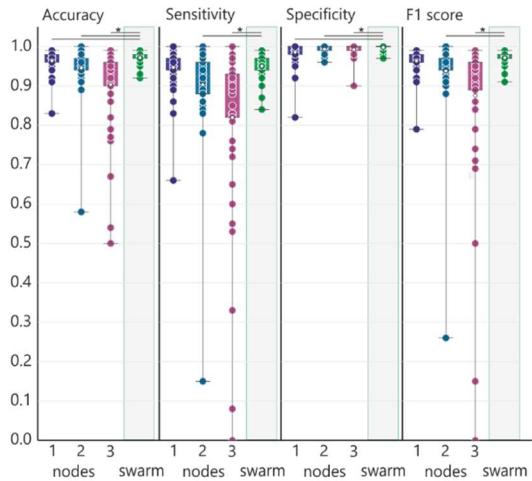
668 **(a-l)** Principle of distribution of data to individual Swarm Edge Nodes (short ‘nodes’). Nodes
669 1-3 were used for training, node 4 for testing. Swarm Learning (SL) was achieved by
670 integrating nodes 1-3 for training following procedures described in detail in Supplementary
671 Information. All scenarios use dataset B and use acute TB as case and the remaining samples
672 as controls. Left panels **(a,c,e,g,i,k)** illustrate the setting of the scenarios concerning
673 distribution of cases (red bar) and controls (blue bar) to individual nodes, as well as total
674 number of samples used for the scenario. Percentage at each node reflects the use of samples
675 out of the complete dataset. **(a)** Scenario with even number of cases at each training node
676 and the test node. **(b)** Evaluation of the scenario presented in (a) showing accuracy, sensitivity
677 and specificity of five permutations for each training node and SL at node 4 (test node) as box-
678 whisker plot (mean, 1st and 3rd quartile, whisker type Min/Max). **(c)** Scenario similar to (a) but
679 with six training nodes. **(d)** Evaluation of scenario (c) as described in (b) but for all six training
680 nodes. **(e)** Scenario where the training nodes have evenly distributed numbers of cases and
681 controls at each training node, but node 2 has lower numbers of samples. **(f)** Evaluation of
682 scenario (e) as described in (b). **(g)** Scenario similar to (e) but with reduced prevalence at the
683 test node. **(h)** Evaluation of scenario (g) as described in (b). **(i)** Scenario with even distribution
684 of cases and controls at each training node, but node 1 only has a very small training set. The
685 test set is evenly distributed. **(j)** Evaluation of scenario (i) as described in (b). **(k)** Scenario
686 similar to (i) but with uneven distribution in the test node. **(l)** Evaluation of scenario (k) as
687 described in (b). Statistical differences between results derived by SL and individual nodes
688 including all permutations performed were calculated with Wilcoxon signed rank test with
689 continuity correction; asterisk and line: p<0.05.

Figure 4

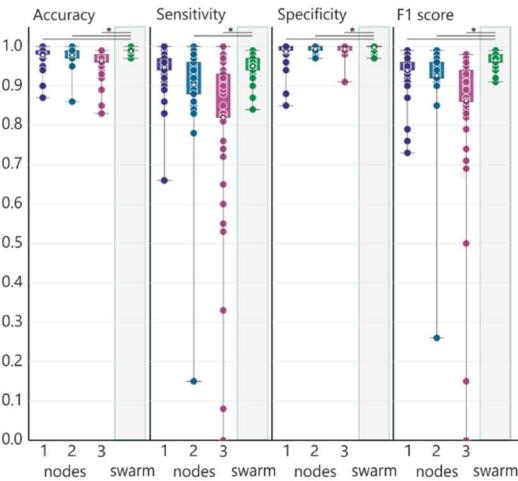
a



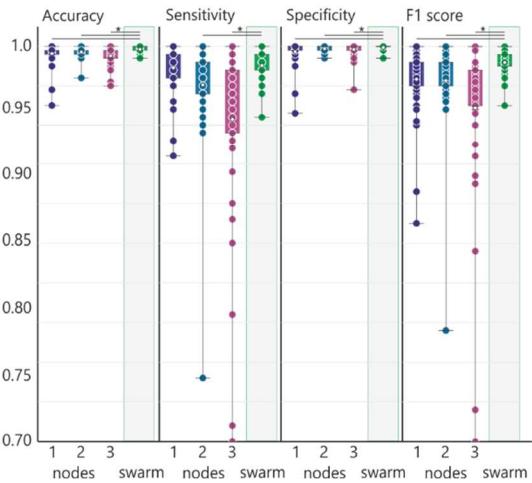
b



c



d



690

691

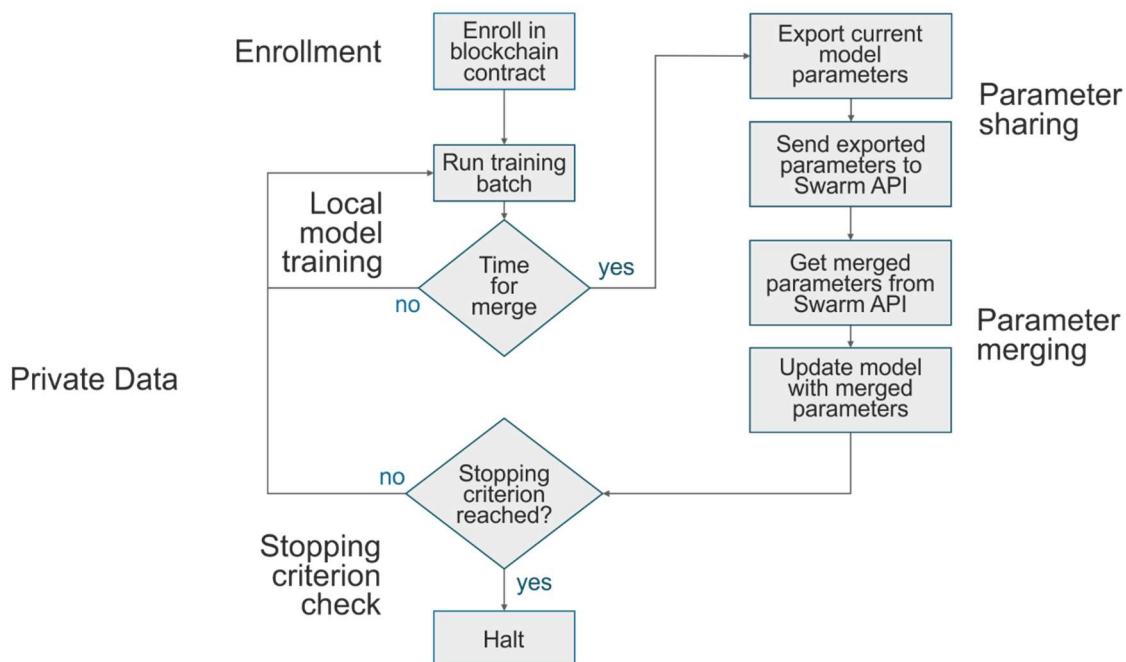
692 **Figure 4. Identification of COVID-19 patients in an outbreak scenario**

693 **(a)** Description of an outbreak scenario for COVID-19 using Dataset C. Nodes 1-3 were used
694 for training, node 4 for testing. Swarm Learning (SL) was achieved by integrating nodes 1-3
695 for training following procedures described in detail in Supplementary Information. COVID-19
696 samples were used as cases. In this scenario, node 1 would be the outbreak node with the
697 highest prevalence. Training node 2 has fewer cases and is an early secondary node, and
698 node 3 acts as a later secondary node. The spreading is tested on the testing node with three
699 different prevalences **(b,c,d)** and shown as box-whisker plot (mean, 1st and 3rd quartile,
700 whisker type Min/Max). **(b)** Evaluation of (a) with even prevalence showing accuracy,
701 sensitivity, specificity and F1-score of fifty permutations for each training node and the SL
702 (node 4). **(c)** Evaluation (as described in (b)) of scenario (a) using a 1:2 ratio for cases and
703 controls in the test set. **(d)** Evaluation (as described in (b)) of scenario (a) using a 1:10 ratio in
704 the test set to simulate detection in regions with new infections. Statistical differences between
705 results derived by SL and individual nodes including all permutations performed were
706 calculated with Wilcoxon signed rank test with continuity correction; asterisk and line: p<0.05.

707

Extended Data Figure 1

a



708

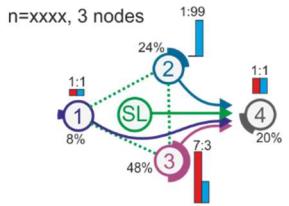
709

710 Extended Data Figure 1. corresponding to Fig. 1

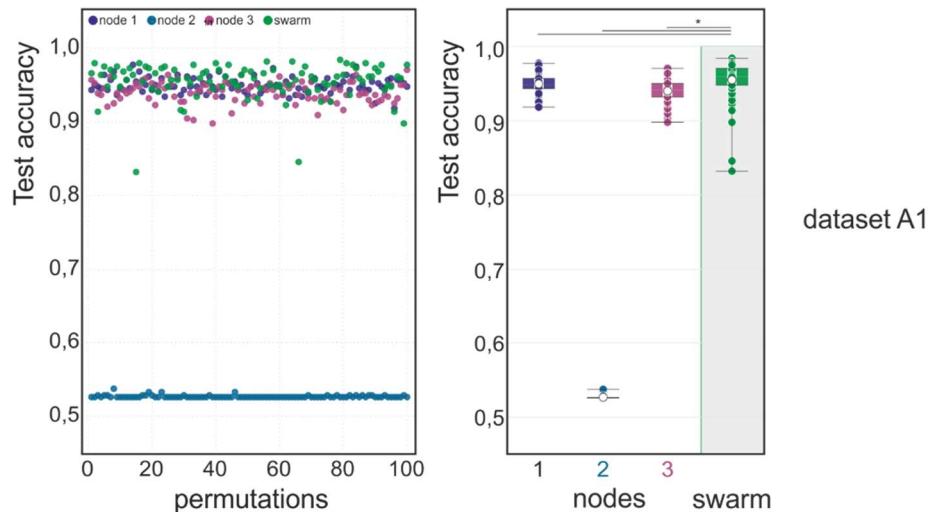
711 Schematics of the principles of the workflow of Swarm Learning once the nodes have been
712 enrolled within the Swarm Network via private permissioned blockchain contract.

Extended Data Figure 2

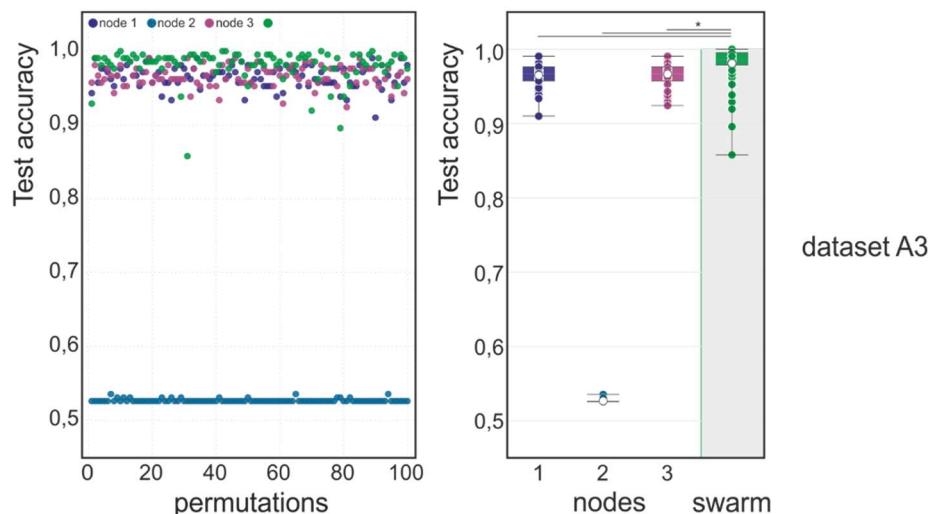
a



b



c



713

714

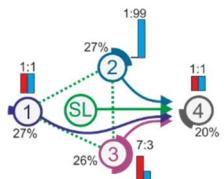
715 **Extended Data Figure 2. Scenario corresponding to Fig. 2b in dataset A1 and A3**

716 Main settings are identical to what is described in Fig. 2 for Dataset A2. **(a)** Scenario with
717 different prevalence of AML and different number of samples at each training node. The test
718 set has an even distribution. **(b)** Evaluation of test accuracy for 100 permutations of dataset
719 A1 per node and swarm. **(c)** Evaluation using dataset A3. Statistical differences between
720 results derived by SL and individual nodes including all permutations performed were
721 calculated with Wilcoxon signed rank test with continuity correction; asterisk and line: p<0.05.

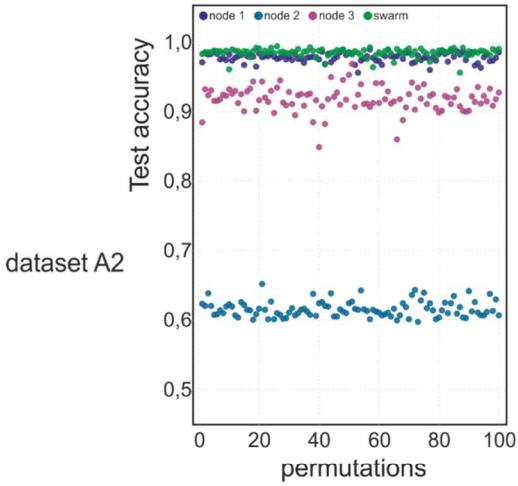
Extended Data Figure 3

a

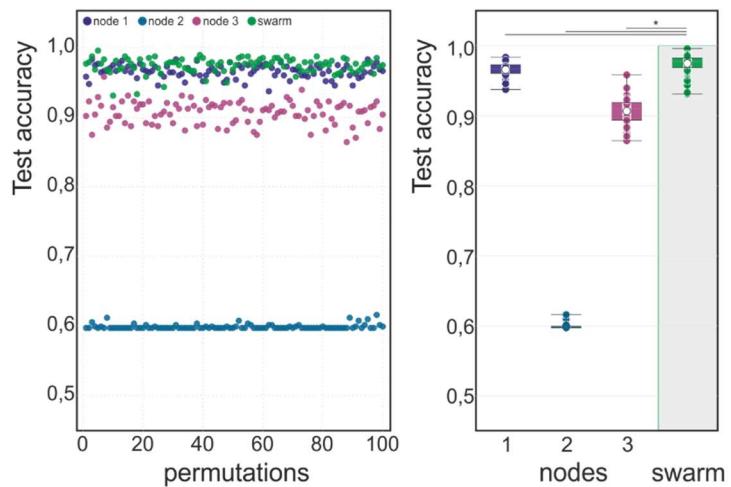
Dataset A1, n=2398, 3 nodes
Dataset A3, n=1008, 3 nodes



b

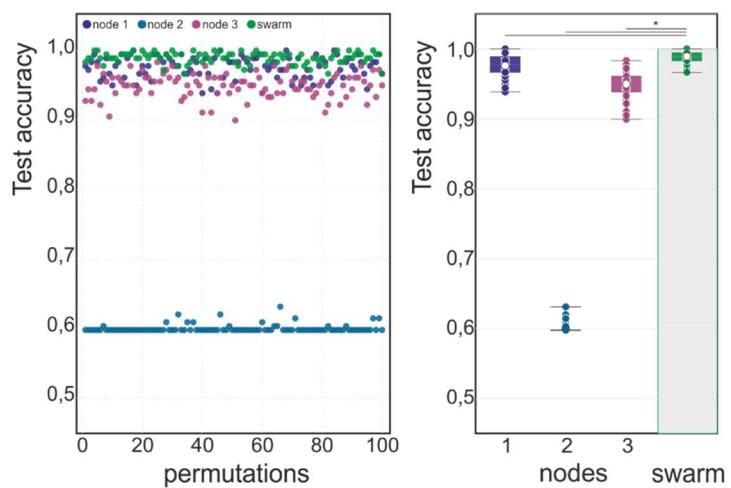


c



dataset A1

d



dataset A3

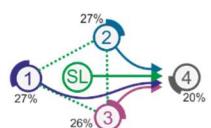
723 **Extended Data Figure 3. Scenario corresponding to Fig. 2c in dataset A1 and A3**

724 Main settings are identical to what is described in Fig. 2 for dataset A2. **(a)** Scenario with
725 similar training set sizes per node but decreasing prevalence. The test set ratio is 1:1. **(b)**
726 Evaluation of the test accuracy over 100 permutation for dataset A2 (corresponding to Fig.
727 2c). **(c)** Evaluation of the test accuracy over 100 permutation for dataset A1. **(d)** Evaluation of
728 the test accuracy over 100 permutation for dataset A3. Box-whisker plots (mean, 1st and 3rd
729 quartile, whisker type Min/Max). Statistical differences between results derived by SL and
730 individual nodes including all permutations performed were calculated with Wilcoxon signed
731 rank test with continuity correction; asterisk and line: p<0.05.

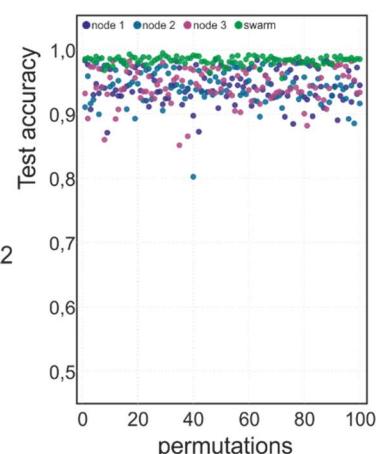
Extended Data Figure 4

a

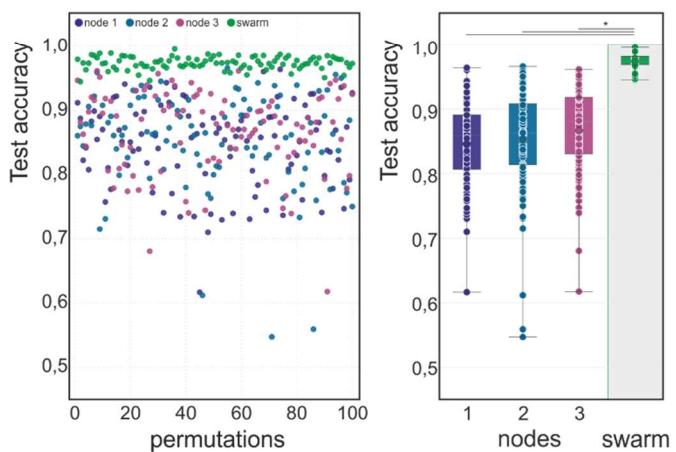
Dataset A1, n=2311, 3 nodes
Dataset A3, n=1083, 3 nodes
varying ratios of cases and controls



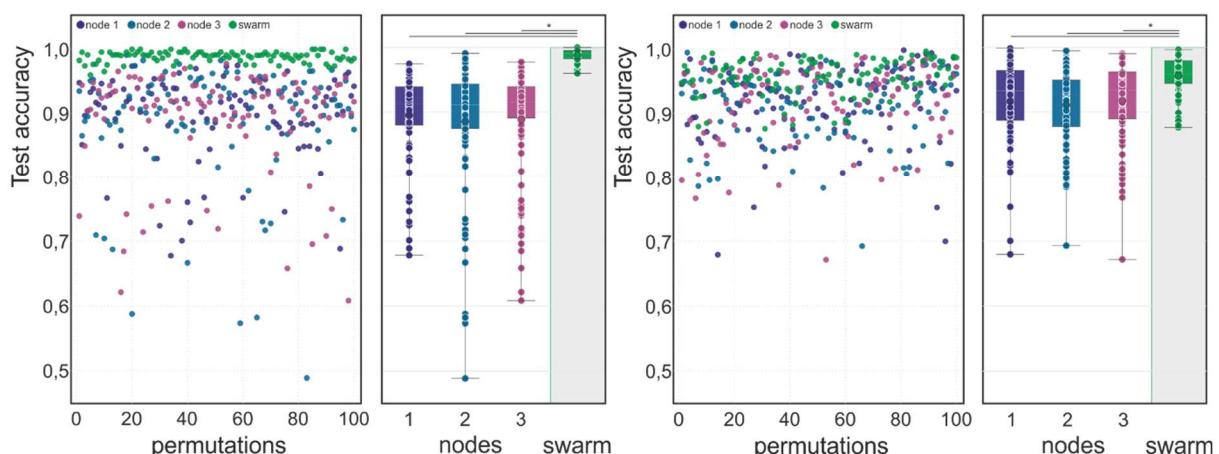
b



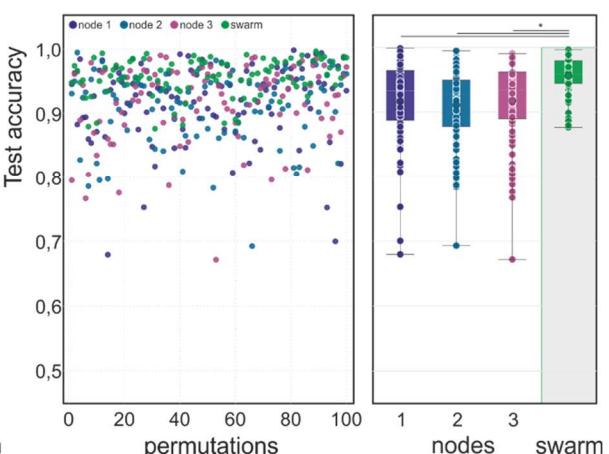
c



d



e



732

dataset A3

dataset A2

733

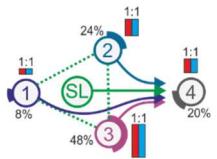
734 **Extended Data Figure 4. Scenario corresponding to Fig. 2d in dataset A1 and A3**

735 Main settings are identical to what is described in Fig. 2 for dataset A2. (a) Scenario with
736 similar sample sizes among three nodes, but with independent studies at each training node.
737 Case and control ratios varied for each permutation. Testing samples are sampled from the
738 studies also present in the training data. (b) Evaluation of the test accuracy over 100
739 permutation for dataset A2 (corresponding to Fig. 2d). (c) Evaluation of the test accuracy over
740 100 permutation for dataset A1. (d) Evaluation of the test accuracy over 100 permutation for
741 dataset A3. (e) In this scenario, samples at the test node were derived from published studies
742 completely independent from the studies used for training at the training nodes. Evaluation of
743 the test accuracy over 100 permutation for dataset A2. Box-whisker plots (mean, 1st and 3rd
744 quartile, whisker type Min/Max). Statistical differences between results derived by SL and
745 individual nodes including all permutations performed were calculated with Wilcoxon signed
746 rank test with continuity correction; asterisk and line: p<0.05.

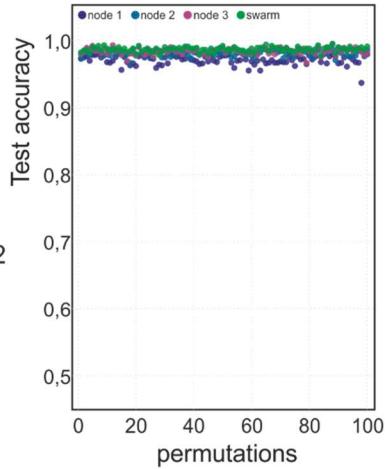
Extended Data Figure 5

a

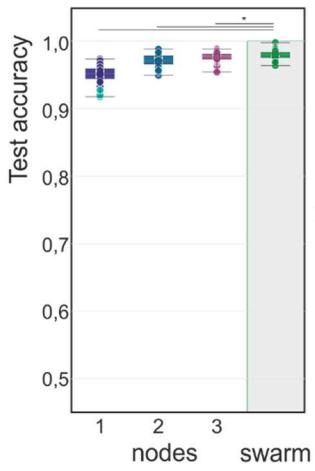
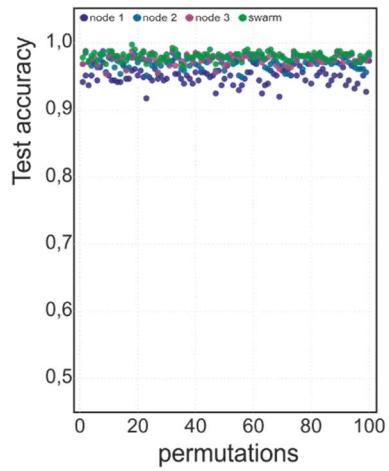
Dataset A1, n=2074, 3 nodes
Dataset A3, n=1008, 3 nodes



b

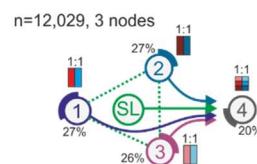


c

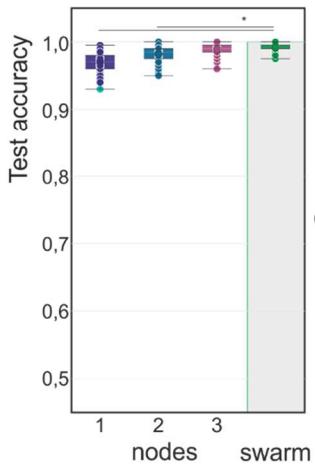
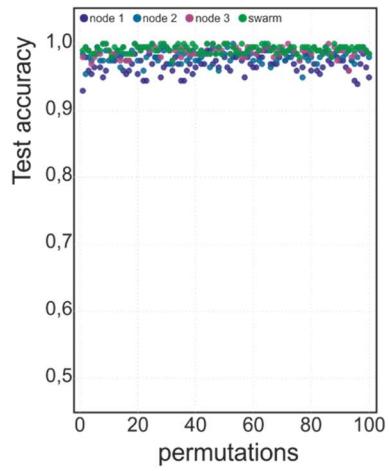


dataset A1

e

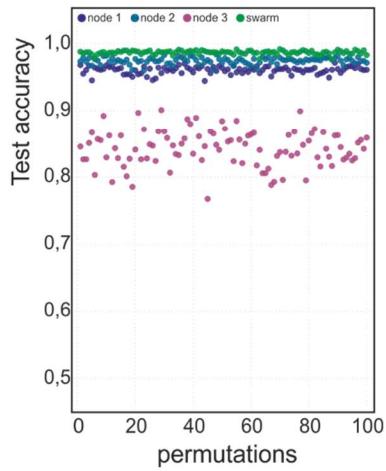


d



dataset A3

f



747

748

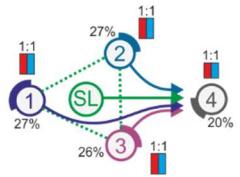
749 **Extended Data Figure 5. Scenario corresponding to Fig. 2e in dataset A1 and A3**

750 Main settings are identical to what is described in Fig. 2 for dataset A2. **(a)** The case:control
751 distribution is even, the training sets increase from node 1 to node 3. The test set is evenly
752 split. **(b)** Test accuracy for evaluation of dataset A2 (corresponding to Fig. 2e). **(c)** Test
753 accuracy for evaluation of dataset A1. **(d)** Test accuracy for evaluation of dataset A3. **(e)**
754 Scenario where the data sets A1, A2, and A3 are assigned to a single training node each.
755 Scenario similar to (a) but with equal training set sizes. **(f)** Evaluation results of 100
756 permutations (corresponding to Fig. 2f). Box-whisker plots (mean, 1st and 3rd quartile, whisker
757 type Min/Max). Statistical differences between results derived by SL and individual nodes
758 including all permutations performed were calculated with Wilcoxon signed rank test with
759 continuity correction; asterisk and line: p<0.05.

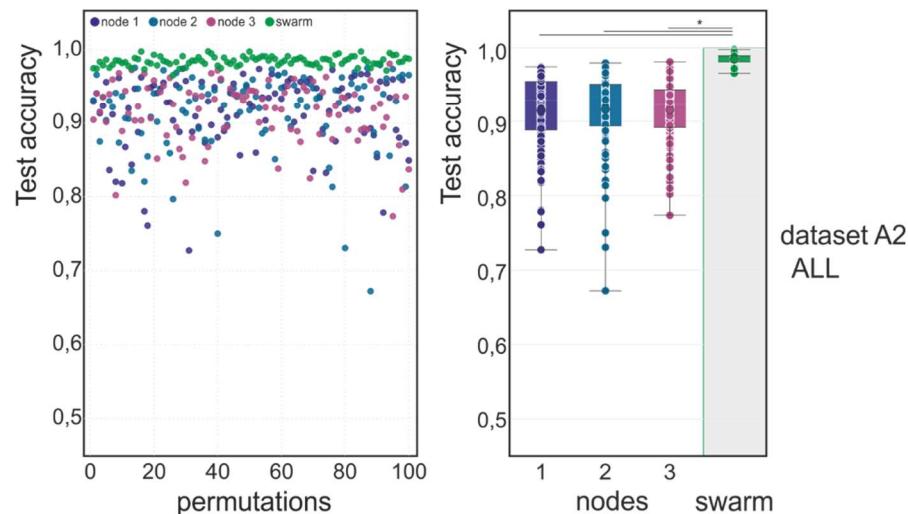
Extended Data Figure 6

a

Dataset A2, n=2318, 3 nodes



b



760

761

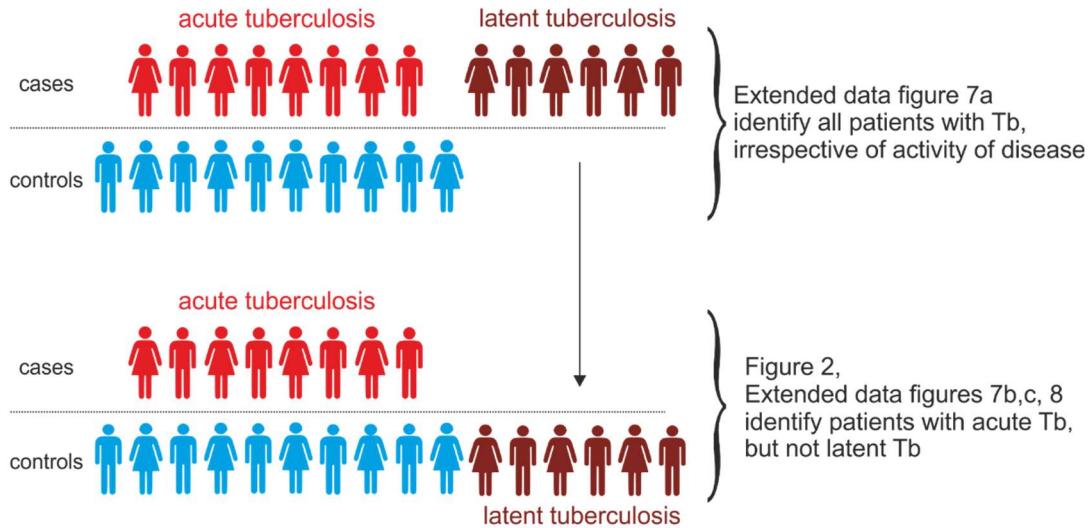
762 **Extended Data Figure 6. Scenario for ALL in dataset 2**

763 Main settings are identical to what is described in Fig. 2 for dataset A2. Here cases are
764 samples derived from patients with ALL, while all other samples are controls (including AML).
765 **(a)** Scenario for the detection of ALL in dataset A2. The training sets are evenly distributed
766 among the nodes. The test ratio is 1:1. **(b)** Evaluation of scenario (a) for test accuracy over
767 100 permutations. Box-whisker plot (mean, 1st and 3rd quartile, whisker type Min/Max).
768 Statistical differences between results derived by SL and individual nodes including all
769 permutations performed were calculated with Wilcoxon signed rank test with continuity
770 correction; asterisk and line: p<0.05.

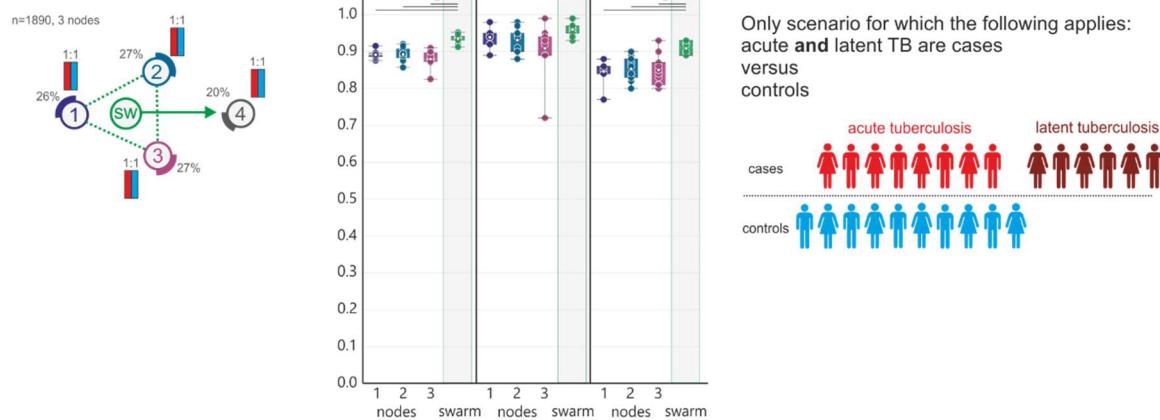
Extended Data Figure 7

a

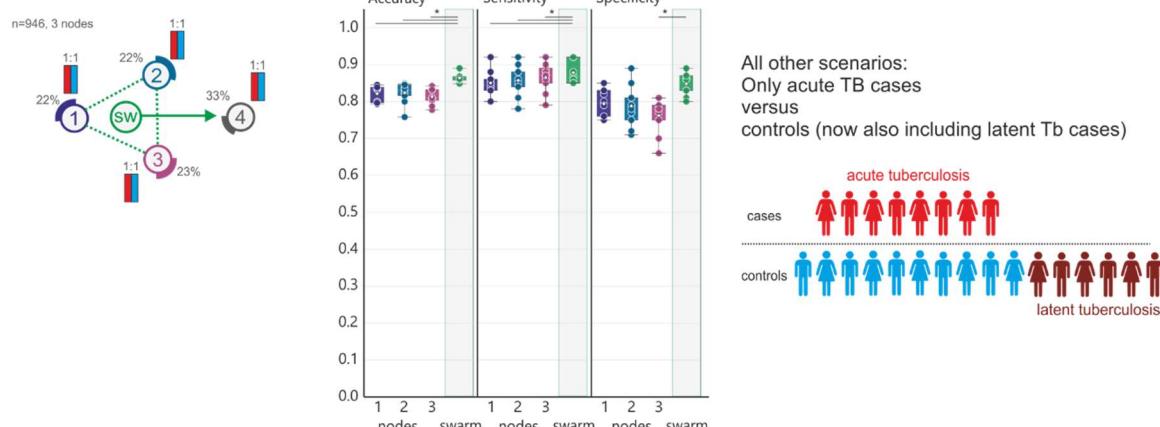
group settings in use case 2 (tuberculosis)



b



c

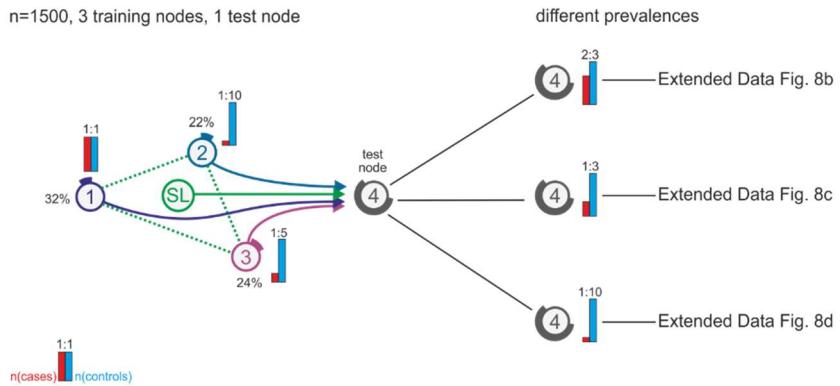


772 **Extended Data Figure 7. Scenario for detecting all Tb versus controls**

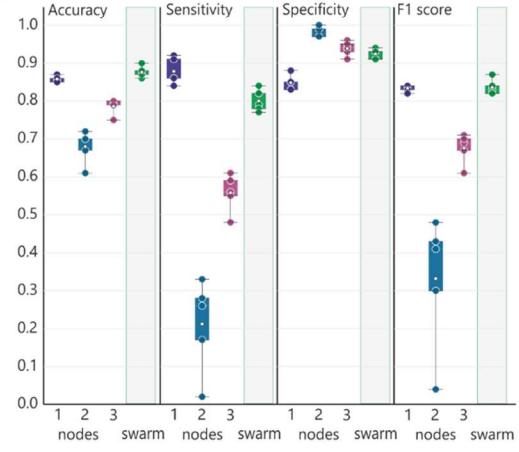
773 **(a)** Description of the different group settings used based on the assignment of latent Tb to
774 control or case. **(b)** Evaluation of a scenario where acute and latent Tb are cases. The data is
775 evenly distributed among the training nodes. The scenario is evaluated as described in Figure
776 3 (b). **(c)** Scenario designed similar to (b) but latent Tb is part of control. Box-whisker plot
777 (mean, 1st and 3rd quartile, whisker type Min/Max). Statistical differences between results
778 derived by SL and individual nodes including all permutations performed were calculated with
779 Wilcoxon signed rank test with continuity correction; asterisk and line: p<0.05.

Extended Data Figure 8

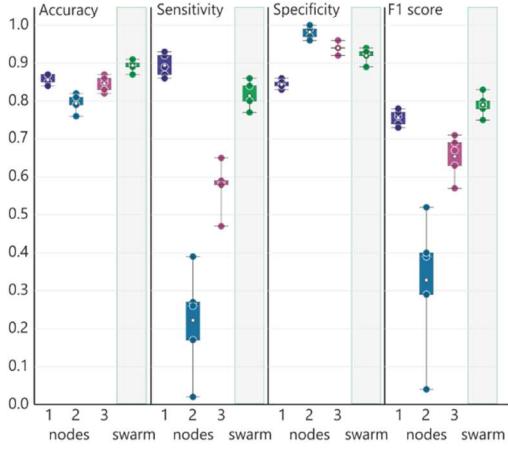
a



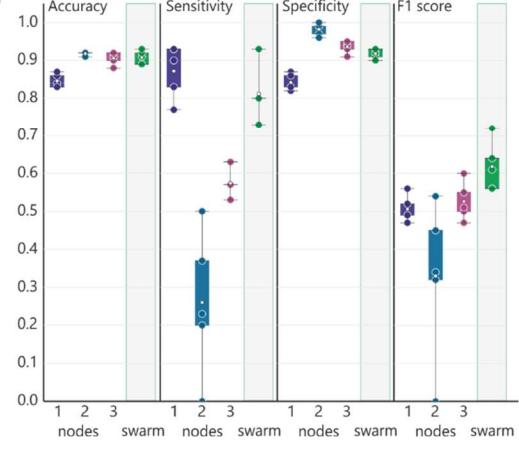
b



c



d



780

781

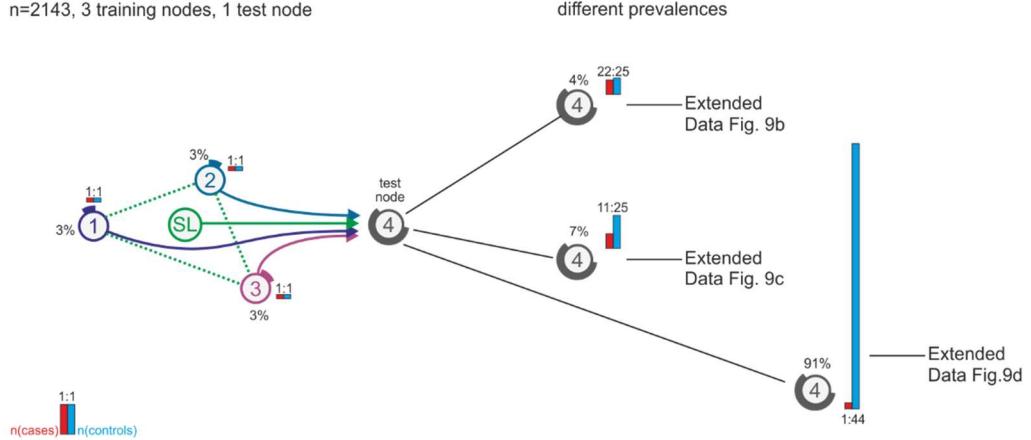
782 **Extended Data Figure 8: Scenario detecting acute Tb with low prevalence at training**
783 **nodes**

784 **(a)** Scenario with training nodes having different prevalence: node 2 has only a 1:10 ratio.
785 Three prevalence scenarios are used in the test set. **(b)** Evaluation of scenario (a) showing
786 accuracy, sensitivity, specificity and F1 score. **(c)** Similar scenario as in (a) but prevalence
787 changed to 1:3 cases: controls in the training set. **(d)** Similar scenario as in (a) but prevalence
788 changed to 1:10 cases: controls in the training set. Box-whisker plot (mean, 1st and 3rd
789 quartile, whisker type Min/Max). Statistical differences between results derived by SL and
790 individual nodes including all permutations performed were calculated with Wilcoxon signed
791 rank test with continuity correction; asterisk and line: p<0.05.

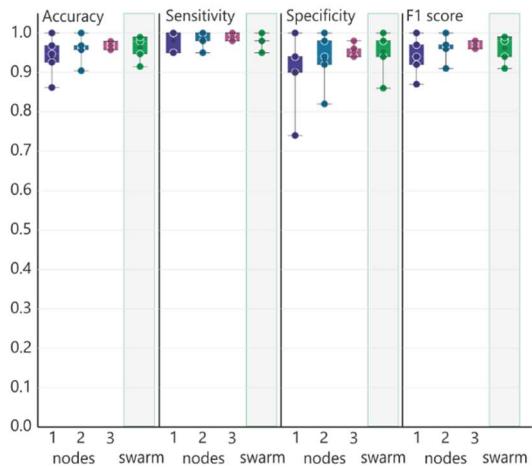
Extended Data Figure 9

a

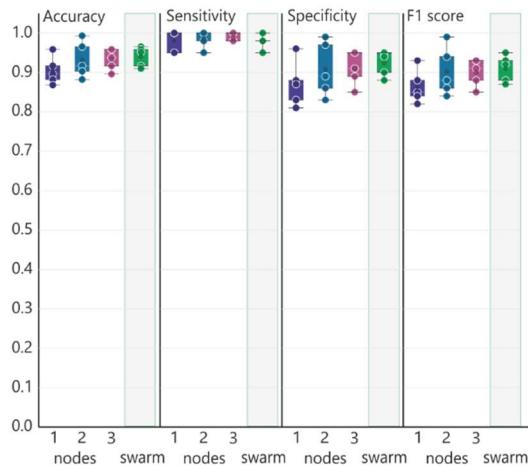
n=2143, 3 training nodes, 1 test node



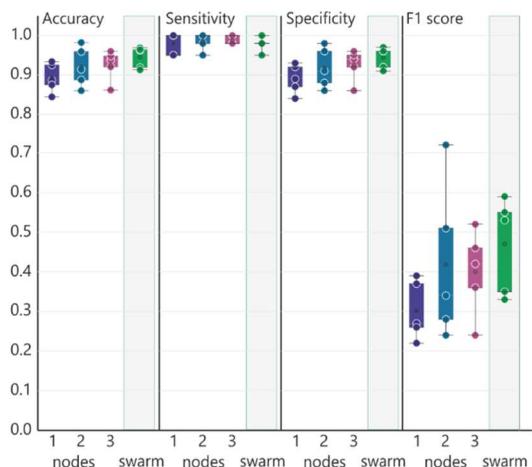
b



c



d



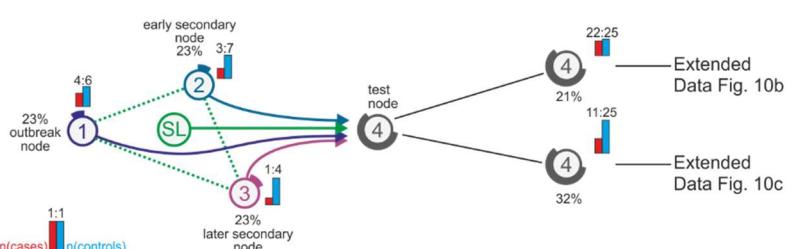
794 **Extended Data Figure 9. Baseline scenario for detecting COVID-19 patients**

795 (a) Scenario with even training set distribution among nodes 1-3. Three different testing sets
796 with different prevalence are simulated. **(b)** Evaluation of (a) for a 22:25 case: control ratio
797 showing accuracy, sensitivity, specificity and F1 score. **(c)** Evaluation results of scenario (a)
798 for a 11:25 ratio. **(d)** Evaluation results of scenario (a) for a 1:44 prevalence. Box-whisker plot
799 (mean, 1st and 3rd quartile, whisker type Min/Max). Statistical differences between results
800 derived by SL and individual nodes including all permutations performed were calculated with
801 Wilcoxon signed rank test with continuity correction; asterisk and line: p<0.05.

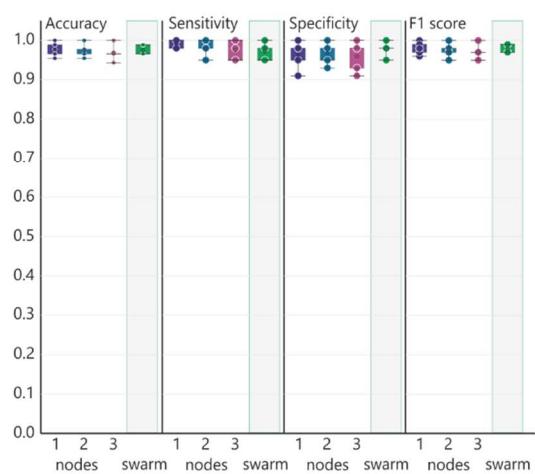
Extended Data Figure 10

a

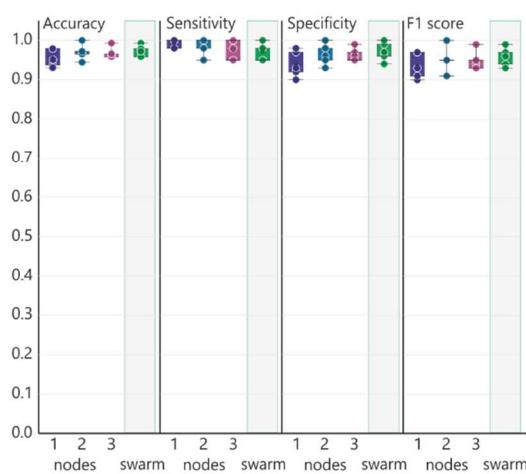
n=444, 3 training nodes, 1 test node



b

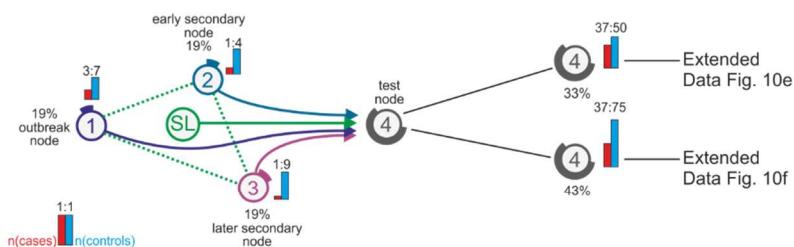


c

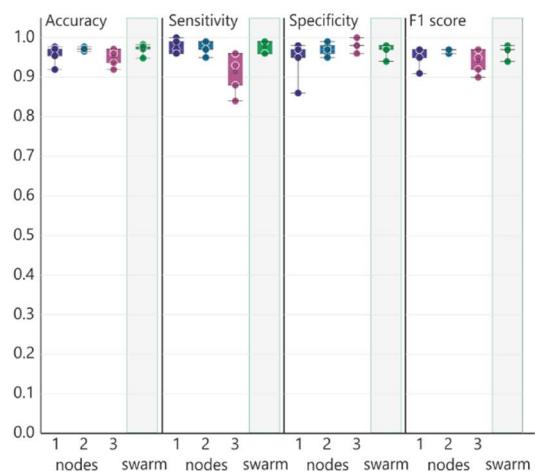


d

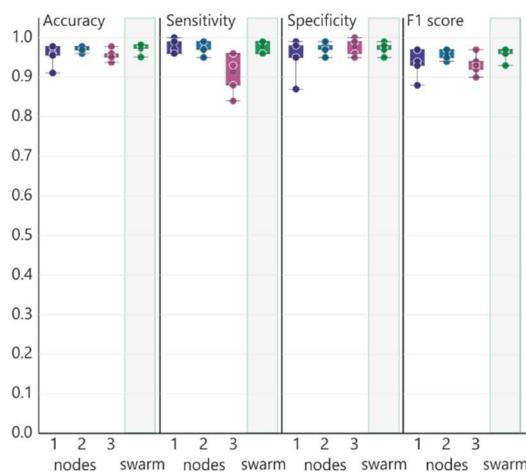
n=474, 3 training nodes, 1 test node



e



f



803 **Extended Data Figure 10. Scenario with reduced prevalence at training nodes for**
804 **detecting COVID-19 patients**

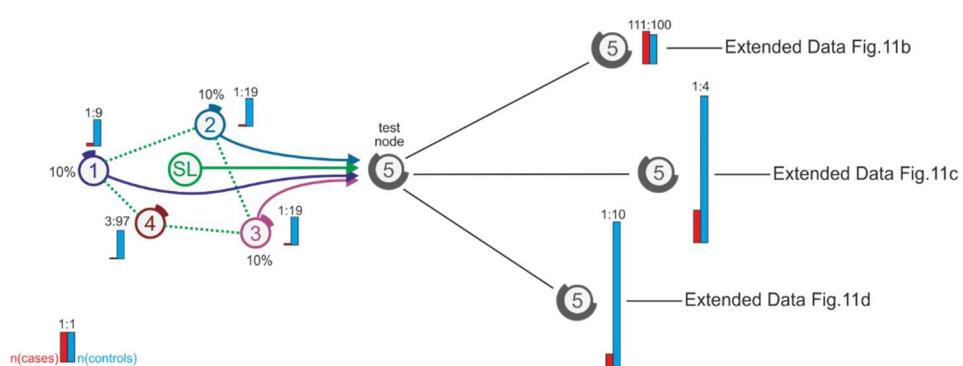
805 **(a)** This scenario has the same sample size at each training node, but the prevalence
806 decreases from node 1 to node 3. There are two different test sets (b) and (c). **(b)** Evaluation
807 of scenario (a) with 22:25 ratio at the test node. **(c)** Results for the evaluation of scenario (a)
808 with reduced prevalence. **(d)** Scenario similar to (a) but the prevalence has a steeper decrease
809 between node 1 and 3. **(e)** Evaluation of scenario (d) with a ratio of 37:50 at the test node. **(f)**
810 Evaluation of (d) with a reduced prevalence compared to (e). Box-whisker plot (mean, 1st and
811 3rd quartile, whisker type Min/Max). Statistical differences between results derived by SL and
812 individual nodes including all permutations performed were calculated with Wilcoxon signed
813 rank test with continuity correction; asterisk and line: p<0.05.

Extended Data Figure 11

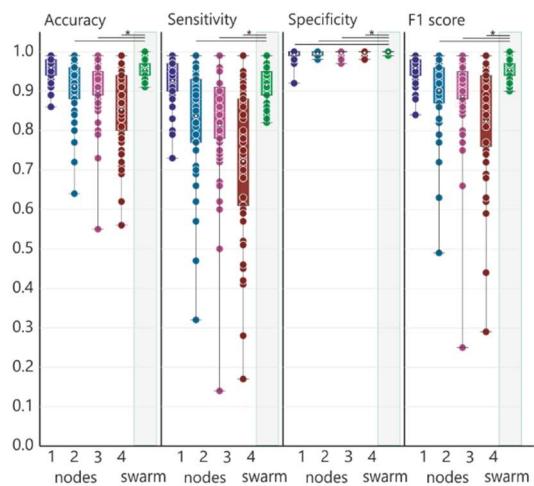
a

n=899, 3 training nodes, 1 test node

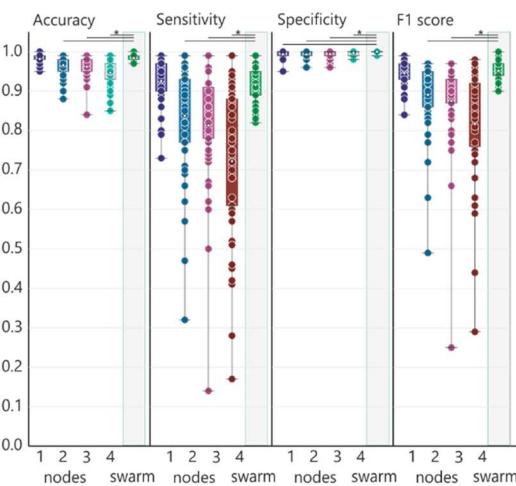
different prevalences



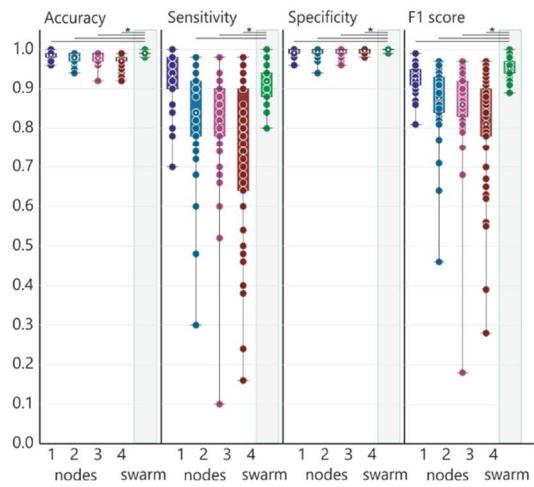
b



c



d



816 **Extended Data Figure 11. Scenario with reduced prevalence in training and test set at**
817 **a 4-node setting (a)** This scenario has even training set sizes among the nodes with the
818 prevalence ranging from 10% at node 1 to 3% at nodes 3 and 4. There are three different test
819 sets **(b), (c) and (d)** with decreasing prevalence and increasing total sample size. **(b)**
820 Evaluation of scenario **(a)** with 111:100 ratio. **(c)** Evaluation of scenario **(a)** with 1:4 ratio and
821 increased sample number of the test set. **(d)** Results of scenario **(a)** with 1:10 prevalence and
822 increased sample number of the test set. Box-whisker plot (mean, 1st and 3rd quartile, whisker
823 type Min/Max). Statistical differences between results derived by SL and individual nodes
824 including all permutations performed were calculated with Wilcoxon signed rank test with
825 continuity correction; asterisk and line: p<0.05.

826 **Supplementary Information**

827 **(Material and Methods)**

828

829 **Datasets**

830 **Peripheral blood mononuclear cell (PBMC) derived transcriptome dataset (Dataset A)**

831 We used a previously described dataset containing over 12,000 transcriptomes derived from
832 peripheral blood mononuclear cells (PBMC), deposited at the National Center for
833 Biotechnology Information Gene Expression Omnibus⁶⁸ (GEO) under SuperSeries
834 GSE122517 or via the individual SubSeries GSE122505 (dataset 1), GSE122511 (dataset 2)
835 and GSE122515 (dataset 3). Briefly, this dataset was generated by inspection of all publicly
836 available datasets at GEO on September 20th, 2017. Inclusion criteria were cell type (PMBCs)
837 and species (*Homo sapiens*). Existing GEO SuperSeries were excluded to avoid duplicated
838 samples. According to data generation method, three datasets were established; dataset 1,
839 generated with Affymetrix HG-U133 A microarrays (n=2,500), dataset 2 with Affymetrix HG-
840 U133 2.0 microarrays (n=8,348), and dataset 3 with high-throughput RNA sequencing (RNA-
841 seq)(n=1,181). Data were curated as previously described⁴⁷. All sample information is listed
842 in **Supplementary Table 2**.

843

844 **Whole blood derived transcriptomes for the prediction of tuberculosis (Dataset B)**

845 To establish a dataset based on whole blood transcriptomes we generated new data from
846 healthy controls (Rhineland Study) and combined these with previously generated data that
847 had been deposited in Gene Expression Omnibus (GEO). We screened for transcriptome
848 datasets derived from human whole blood samples, which were collected using the PAXgene
849 Blood RNA System. In total, nine independent datasets were selected to be included in the
850 present study (GSE101705 (n=44); GSE107104 (n=33), GSE112087 (n=120), GSE128078
851 (n=99), GSE66573 (n=14), GSE79362 (n=355), GSE84076 (n=36); GSE89403 (n=914)). The
852 newly generated 384 whole blood samples were sampled in context of the Rhineland Study
853 led by the German Center for Neurodegenerative Diseases (DZNE), which is an extensive
854 longitudinal study monitoring healthy individuals over 2 decades. Approval to undertake the
855 Rhineland Study was obtained from the ethics committee of the University of Bonn, Medical
856 Faculty. The study is carried out in accordance with the recommendations of the International
857 Conference on Harmonization (ICH) Good Clinical Practice (GCP) standards (ICH-GCP).
858 Written informed consent was obtained from all participants in accordance with the Declaration

859 of Helsinki. Overnight fasting blood was collected from all participants, including a PAXgene®
860 tube for RNA extraction and RNA-seq analysis. In total, Dataset B contained 1999 samples
861 from patients with active tuberculosis (n=775), latent tuberculosis (n=277), fatigue (n=55),
862 autoimmune diseases (n=68), HIV (n=16) and controls (n=808). Sample information is listed
863 in **Supplementary Table 2**.

864

865 **Whole blood derived transcriptome dataset for the prediction of COVID-19 (Dataset C)**

866

867 To develop classifiers based on whole blood transcriptomes able to predict COVID-19 patients
868 we collected an additional 134 PAXgene® tubes for RNA extraction and RNA-seq analysis
869 from COVID-19 patients, of which 93 whole blood samples at the Intensive Care Unit of the
870 Radboud University Medical Centre in Nijmegen, the Netherlands, and 41 samples were either
871 collected at the Sotiria Athens General Hospital or the ATTIKON University General Hospital
872 in Athens, Greece. For all COVID-19 patients, the study was carried out in accordance with
873 the applicable rules concerning the review of research ethics committees and informed
874 consent. All patients or legal representatives were informed about the study details and could
875 decline to participate. COVID-19 was diagnosed by a positive SARS-CoV-2 RT-PCR test in
876 nasopharyngeal or throat swabs and/or by typical chest CT-scan finding. Blood for RNA-seq
877 analysis was sampled on day 0 to 11 after admission. In the cohort in Athens, blood samples
878 from ten healthy donors who were tested negative on SARS-CoV-2 were included as controls.
879 The newly generated samples from the COVID-19 patients and the controls from Athens were
880 combined with dataset B (see above) to establish Dataset C. As a result, in addition to the
881 1999 samples derived from Dataset B, Dataset C included further 10 healthy controls and 134
882 dutch COVID-19 samples, which makes a total of 2,143 samples. Sample information is listed
883 in **Supplementary Tables 2 and 6**.

884

885 **Pre-processing**

886 **PBMC transcriptome dataset (Dataset A)**

887 We used a previously published dataset compiled for predicting AML in blood transcriptomes
888 derived from peripheral blood mononuclear cells (PBMC)⁴⁷. Briefly, all raw data files were
889 downloaded from GEO and the RNA-seq data was preprocessed using the kallisto aligner
890 against the human reference genome gencode v27 (GRCh38.p10). For normalization, we
891 considered all platforms independently, meaning that normalization was performed separately
892 for the samples in Dataset A1, A2 and A3, respectively. Microarray data (Datasets A1 and A2)

893 was normalized using the robust multichip average (RMA) expression measures⁶⁹, as
894 implemented in the R package affy⁷⁰. RNA-seq data (Dataset A3) was normalized with the R
895 package DESeq2 using standard parameters⁷¹. In order to keep the datasets comparable,
896 data was filtered for genes annotated in all three datasets, which resulted in 12,708 genes. No
897 filtering of low-expressed genes was performed. All scripts used in this study for pre-
898 processing are provided as a docker container on Docker Hub (docker hub,
899 https://hub.docker.com/r/schultzelab /aml_classifier).

900

901 **Whole blood derived transcriptome datasets (Datasets B and C)**

902 Since alignment of whole blood transcriptome data can be performed in numerous different
903 ways, we re-aligned all downloaded and collected datasets which were 4.7 Terabyte in size
904 and comprised a total of 7.8 Terabases, to the human reference genome gencode v33
905 (GRCh38.p13) and quantified transcript counts using STAR, an ultrafast universal RNA-seq
906 aligner (version 2.7.3a)⁷². For all samples in Datasets B and C, raw counts were imported
907 using DESeqDataSetFromMatrix function and size factors for normalization were calculated
908 using the DESeq function using standard parameters⁷¹. This was done separately for Dataset
909 B and Dataset C. Since some of the samples were prepared with poly-A selection to enrich
910 for protein-coding mRNAs, we filtered the complete dataset for protein-coding genes in order
911 to ensure greater comparability across library preparation protocols. Furthermore, we
912 excluded all ribosomal protein-coding genes, as well as mitochondrial genes and genes coding
913 for hemoglobins, which resulted in 18,135 transcripts as the feature space in Dataset B and
914 19,358 transcripts in Dataset C. Furthermore, transcripts with an overall expression < 10 were
915 excluded from further analysis. Other than that, no filtering of transcripts was performed. Prior
916 to use in machine learning we performed a rank transformation to normality on both datasets
917 B and C⁷³. Briefly, transcript expression values were transformed from RNAseq counts to their
918 respective ranks. This was done transcript-wise, meaning all transcript expression values per
919 sample were given a rank based on ordering them from lowest to highest value. The rankings
920 were then turned into quantiles and transformed via the inverse cumulative distribution
921 function of the Normal distribution. This leads to all transcripts following the exact same
922 distribution (that is, a standard Normal with a mean of 0 and a standard deviation of 1) across
923 all samples

924 **Methods details**

925 **Scenarios for prediction of AML**

926 We previously demonstrated that ML on PBMC transcriptomes can be utilized to predict
927 AML⁴⁷. Based on this experience, we generated sample sets within three independent
928 transcriptome datasets (dataset A1-A3, see above) to assess different scenarios in a three-
929 node setting for training with a fourth node only used for testing. As indicated in **Fig. 2**, six
930 scenarios with varying numbers of samples per node and varying ratios between cases and
931 controls at each node were defined. For predicting AML, all samples derived from AML
932 patients were classified as cases, while all other samples were labeled controls. When
933 predicting ALL, all samples derived from ALL patients were classified as cases and all others
934 as controls. For each scenario (**Fig. 2**) and each dataset we permuted the sample distribution
935 100 times, resulting in a total of 5,594 individual predictions. The different scenarios were
936 chosen to address the influence of sample numbers per node, the case control ratio, study
937 design-related batch effects, and transcriptome technologies used on classifier performance
938 at the nodes, but more importantly on swarm learning performance. Sample distributions for
939 all permutations within all scenarios are listed in **Supplementary Table 1**.

940

941 Scenarios for detecting patients with acute TB

942 In line with the experience we gained from the prediction of AML, we used dataset B to
943 generate scenarios for the prediction of tuberculosis in various settings, again using different
944 scenarios in a three-node setting for training with a fourth node only used for testing. In one
945 scenario, all patients with tuberculosis (Tb) including patients with latent and acute Tb were
946 treated as cases, while all others were defined as controls (**Extended Data Fig. 6b**). In all
947 other scenarios, cases were restricted to acute Tb patients' samples, while patients with latent
948 Tb were defined as controls together with all other non-Tb samples. Here, the question to be
949 answered is, whether the classifiers can identify patients with acute Tb and can distinguish
950 them from latent Tb and other conditions.

951 In one scenario (**Fig. 3c-d**), we added three additional training nodes to test dependency of
952 classifier performance by the number of nodes. As indicated in **Fig. 3**, three scenarios with
953 varying numbers of samples per node and varying ratios between cases and controls at each
954 node were defined. For scenarios described within **Fig. 3e,g** and **Fig. 3i,k**, we tested two
955 prevalence scenarios in the test set. For each scenario (**Fig. 3**) we permuted the sample
956 distribution 5-10 times, resulting in a total of 325 individual predictions. To mimic an outbreak
957 scenario, we reduced cases also at the training nodes to determine the effects on Swarm
958 Learning performance. Sample distributions for all permutations within all scenarios are listed
959 in **Supplementary Table 1**.

960

961 **Simulation of an outbreak scenario to detect COVID-19 patients**

962 Based on the promising results obtained with tuberculosis, we next intended to simulate
963 classifier building and testing for the prediction of COVID-19 in a SL setting. We used dataset
964 B and added 144 additional samples, of which 139 samples were derived from COVID-19
965 patients (see above). We applied a three-node setting for training with a fourth node only used
966 for testing.

967 In one scenario (**Extended Data Fig. 8**), we kept cases (n=30) and controls (n=30) evenly
968 distributed among the three training nodes and tested three different prevalence scenarios at
969 the test node (22:25; 11:25; 1:44). In a second scenario (**Extended Data Fig. 9a-c**) we
970 changed the ratio of cases and controls at each node (node 1: 40:60, node 2: 30:70, node 3:
971 20:80) and tested two prevalence scenarios at the test node (22:25; 11:25). In a third scenario
972 (**Extended Data Fig. 9a-c**) we further reduced the number of cases at the training nodes
973 further (node 1: 30:70, node 2: 20:80, node 3: 10:90) and tested two prevalence scenarios at
974 the test node (37:50; 37:75).

975 Lastly, we tested an outbreak scenario (**Fig. 4**) with very few cases at the outbreak node 1
976 (20:80), an early secondary node (10:90) and a later secondary node (5:95) and three
977 prevalence scenarios at the test node (1:1, 1:2, 1:10), resulting in a total of 220 individual
978 predictions Sample distributions for all permutations within all scenarios are listed in
979 **Supplementary Table 1**.

980

981 **Application layer**

982 The application layer (see also **Fig. 1g**) consists of disease models for which definitions are
983 given, which samples are cases and which samples are controls. For example, if the classifier
984 is supposed to detect all patients with tuberculosis (Tb), the model includes patients with latent
985 and acute tuberculosis as cases and all other samples as controls. However, if only patients
986 with acute tuberculosis are intended to be detected as cases, the model is changed in that
987 cases are now only patient samples derived from patients with acute Tb, while samples from
988 patients with latent Tb are now treated as controls, similar to all other non-Tb samples. The
989 cases and controls used for each scenario are given in the result section in more detail. For
990 each mode, classifiers are generated by applying neural networks (for description see below)

991 **Computation and analysis**

992 **Neural network algorithm**

993 We leveraged a deep neural network with a sequential architecture as implemented in the
994 keras library (Keras, <https://keras.io/>, 2015). Briefly, the neural network consists of one input
995 layer, eight hidden layers and one output layer. The input layer is densely connected and
996 consists of 256 nodes, a rectified linear unit activation function and a dropout rate of 40%.
997 From the first to the eighth hidden layer, nodes are reduced from 1024 to 64 nodes, and all
998 layers contain a rectified linear unit activation function, a kernel regularization with an L2
999 regularization factor of 0.005 and a dropout rate of 30%. The output layer is densely connected
1000 and consists of 1 node and a sigmoid activation function. The model is configured for training
1001 with Adam optimization and to compute the binary cross-entropy loss between true labels and
1002 predicted labels.

1003 The model has been translated from R to Python in order to make it compatible with the swarm
1004 learning library. This model is used for training both the individual nodes as well as swarm
1005 learning. The model is trained over 100 epochs, with varying batch sizes. The batch size of 8,
1006 16, 32, 64 and 128 are used depending on the number of training samples.

1007

1008 **Preparation and adaptation of neural network code to be used in a swarm learning 1009 environment**

1010 A swarm callback is introduced to integrate the model with the Swarm Learning library.
1011 Minimum number of nodes for synchronization, synchronization interval, validation dataset
1012 and batch size are passed as parameters to swarm callback. The swarm call back API is

```
1013 swCallback = SwarmCallback(sync_interval = <number of training batches between syncs>,  
1014 min_peers = <minimum peers>,  
1015 val_data = <validation dataset>,  
1016 val_batch_size = <validation batch size>,  
1017 node_weightage = <relative weightage of node's model weights>)
```

1018 sync_interval specifies the synchronization interval,

1019 min_peers specifies the minimum number of nodes for model synchronization,

1020 val_data specifies the validation data set,

1021 val_batch_size specifies the validation batch size,

1022 model_name specifies the name of the model,

1023 node_weightage specifies the relative weightage to be given to model weights of this node

1024

1025 **Parameter tuning**

1026 For some of the scenarios we tuned model hyperparameters. For some scenarios we also
1027 tuned Swarm Learning parameters to get better performance, for example higher sensitivity.

1028 For AML **Fig. 2e, Extended Data Fig. 2 and Fig. 2f**, dropout rate is reduced to 10% to get
1029 better performance. For AML **Fig. 2b, Extended Data Fig. 1**, dropout rate is reduced to 10%
1030 and increased the Epochs to 300 to get better performance. We also used the adaptive_rv
1031 parameter in the Swarm Learning API to adjust the merge frequency dynamically based on
1032 model convergence to improve the training time. For TB and COVID-19 tests dropout rate is
1033 reduced to 10% for all scenarios. For the TB scenarios in **Extended Data Fig. 7a,b**, the
1034 node_weightage parameter of Swarm Learning callback API is used to give more weightage
1035 to the nodes that have higher case samples.

1036

1037 **Infrastructure layer**

1038 **Description of the hardware architecture applied for simulations**

1039 For all simulations provided in this project we used 2 HPE Apollo 6500 Gen 10 server, each
1040 with 4 Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz, a 3.2 TB hard disk drive, 256 GB RAM,
1041 8 Tesla P100 GPUs, 1GB network interface card for LAN access and infiniBand FDR for high
1042 speed interconnect and networked storage access. The Swarm Network is created with 3
1043 nodes, each node is a docker container with 1 GPU. Multiple experiments were run in parallel
1044 using the above described configuration.

1045 Overall, we performed 6,139 analyses including six scenarios for all three AML datasets, nine
1046 scenarios for Tb and 10 scenarios for COVID-19. We performed 5 to 100 permutations per
1047 scenario, each permutation took approximately 30 minutes, which resulted in a total of 3069,5
1048 compute hours.

1049

1050 **The Swarm learning framework, library, distributed ML and blockchain technologies**

1051 Swarm Learning builds on top of two proven technologies — distributed ML and blockchain.
1052 Distributed ML is leveraged to train a common model across multiple nodes with a subset of
1053 the data located at each node — commonly known as the data parallel paradigm in ML —
1054 though without a central parameter server. Blockchain lends the decentralized control,
1055 scalability, and fault-tolerance aspects to the Swarm Network system to enable the framework
1056 to work beyond the confines of a single enterprise.

1057 The Swarm Learning library is a framework to enable decentralized training of ML models
1058 without sharing the data. The Swarm Learning framework is designed to make it possible for
1059 a set of nodes — each node possessing some training data locally — to train a common ML

1060 model collaboratively without sharing the training data itself. This can be achieved by individual
1061 nodes sharing parameters (weights) derived from training the model on the local data. This
1062 allows nodes to maintain the privacy of their raw data. Importantly, in contrast to many existing
1063 federated learning models, a central parameter server is omitted in Swarm Learning.

1064 The nodes that participate in Swarm Learning, register themselves with the Swarm Network
1065 implicitly using the callback API. Here, the Swarm Network interacts with other peers using
1066 blockchain for sharing parameters and for controlling the training process. On each node, a
1067 simple Swarm callback API has to be used to enable the ML model with Swarm Learning
1068 capacities (see also code presented below). The Swarm container has to be configured to
1069 interact with the Swarm Network (network i/p and port configuration). All other complexities of
1070 setting up network, registration, parameter sharing, and parameter merging are taken care of
1071 by the Swarm callback API and the Swarm Network infrastructure.

1072 Parameters shared from all the nodes are merged to obtain a global model. Moreover, the
1073 merge process is not done by a static central coordinator or parameter server, but rather a
1074 temporary leader chosen dynamically among the nodes is used to perform the merge, thereby
1075 making the Swarm network decentralized. This provides a far greater fault-tolerance than
1076 traditional centralized-parameter-server-based frameworks. All the nodes can perform the role
1077 of training and merging, thereby maximising the usage of local compute. The Swarm Network
1078 implicitly controls this.

1079 The HPE Swarm Learning library contains 2 containers, the Swarm Network container and
1080 the Swarm ML container.

1081 The Swarm Network container includes 1) software to setup and initialize the Swarm Network,
1082 2) management commands to control the Swarm Network, and 3) start/stop Swarm Learning
1083 tasks. This container also encapsulates the blockchain software.

1084 The Swarm ML container includes software to support 1) decentralized training, 2) integration
1085 with ML frameworks, and 3) it exposes APIs for ML models to interact with Swarm Learning.

1086 For any ML model to be applied to Swarm Learning, it needs to be modified using the Swarm
1087 callback API. The callback API provides options to control the Swarm Learning processes. To
1088 convert a ML program into a Swarm ML program the following steps have to be performed:

1089 1. Import the SwarmCallback class from the swarm library

1090 from swarm 'import SwarmCallback'

1091 SwarmCallback is a custom callback class that is built on the Keras Callback class.

1092 2. Instantiate an object of the SwarmCallback class:

```
1093 swarm_callback = SwarmCallback( min_peers = <peer count>,
1094                               sync_interval = <interval>,
1095                               use_adaptive_sync = <bool>,
1096                               val_batch_size = <batch size>,
1097                               val_data = <either a (x_val, y_val) tuple or a
1098                                         generator>
1099                               node_weightage = <relative weightage of node's
1100                                         model weights> ).
```

1101 In this context, ‘min_peers’ specifies the minimum number of network peers required
1102 to synchronize the insights, ‘sync_interval’ specifies the number of batches after which
1103 a synchronization is performed, ‘use_adaptive_sync’ specifies whether the *adaptive*
1104 *sync interval* feature should be used for tuning the sync interval. This feature is turned
1105 off by default; ‘val_batch_size’ specifies the size of each validation batch; ‘val_data’
1106 specifies the validation dataset. It can be either a (x_val, y_val) tuple or a generator;

1107 3. Pass the object to the list of callbacks in Keras training code: model.fit(...,
1108 callbacks = [swarm_callback]). SwarmCallback can be included along with other
1109 callbacks also:

```
1110 es_callback = EarlyStopping(...);
1111 model.fit(..., callbacks = [es_callback, swarm_callback])
```

1112

1113 The Swarm Learning architecture principles

1114 The Swarm Learning framework has two major components, 1) the Swarm ML component
1115 runs a user-defined Machine Learning algorithm, and 2) the Swarm Network component forms
1116 the Swarm Network based on a blockchain network.

1117 The Swarm ML component is implemented as an API available for multiple popular
1118 frameworks such as TensorFlow, Keras, Pytorch. This API provides an interface that is similar
1119 to the training APIs in the native frameworks familiar to data scientists. Calling this API
1120 automatically inserts the required hooks for Swarm Learning so that nodes seamlessly
1121 exchange parameters and subsequently continue the training after setting the local models to
1122 the globally merged parameters. With a few simple code changes, the entire network learns
1123 as one cohort, with all the complexities of control and data flow taking place in an automated
1124 fashion.

1125 Within the Swarm Network component each Swarm ML component interacts with each other
1126 using the Swarm Network component's blockchain platform to maintain global state
1127 information about the model that is being trained and to track the training progress. The Swarm
1128 Network components use this state and progress information to coordinate the working of the
1129 Swarm learning. The Swarm Network is responsible for keeping the decentralized Swarm
1130 network in a globally consistent state. The Swarm Network ensures that all operations and the
1131 corresponding state transitions are performed in a synchronous manner. Both, state and
1132 supported operations of the system are encapsulated in a blockchain smart contract. The
1133 Swarm Network contains the logic to elect the leader of the Swarm for every synchronization,
1134 implement fault-tolerance, and self-healing mechanisms, along with signaling among nodes
1135 for commencement and completion of various phases.

1136 The Swarm Learning framework is designed to run on both commodity and high-end
1137 machines, supporting a heterogeneous set of infrastructure in the network. It can be deployed
1138 within and across data centers.

1139 In contrast to federated learning with star topology and a centralized coordinator, Swarm
1140 Learning can support multiple topologies including fully connected, mesh, star, tree and hybrid
1141 topologies. This flexibility provides multiple options to cater into different use cases.

1142

1143 **The Swarm Learning process**

1144 Swarm Learning provides a callback API to enable swift integration with multiple frameworks.
1145 This API is incorporated into the existing ML code to quickly transform a stand-alone ML node
1146 into a Swarm Learning participant in a non-intrusive way. It offers a set of commands (APIs)
1147 to manage the Swarm Network and control the training.

1148 The Swarm learning process is as follows:

1149 The Swarm Learning process begins with enrollment of nodes with Swarm Network, which is
1150 done implicitly by Swarm callback function when the callback is constructed. During this
1151 process, the relevant attributes of the node are stored in the blockchain ledger. This is a one-
1152 time process.

1153 Nodes will train the local copy of the model iteratively using private data over multiple epochs.
1154 During each epoch, the node trains its local model using one or more data batches for a fixed
1155 number of iterations. It regularly shares its learnings with the other Swarm nodes and
1156 incorporates their insights. Users can control the periodicity of this sharing by defining a

1157 Synchronization Interval in Swarm callback API. This interval specifies the number of training
1158 batches after which the nodes will share their learnings.

1159 At the end of every synchronization interval, when it is time to share the learnings from the
1160 individual models, one of the Swarm nodes is elected as a "leader" using the leader election
1161 logic. This leader node collects the model parameters from each peer node and merges them.
1162 The framework supports multiple merge algorithms such as mean, weighted mean, median,
1163 and so on. Each node then uses these merged parameters to calculate various validation
1164 metrics. These results are compared against the stopping criterion and if it is found to be met,
1165 the Swarm Learning process is halted. Else the nodes use the merged parameters to start the
1166 next training batch.

1167 Swarm Learning library uses blockchain smart contracts to define the leader election logic and
1168 the merge algorithm. The blockchain smart contracts prevents attacks from semi-honest or
1169 dishonest participants.

1170

1171 **Quantification and Statistical Analysis**

1172 We evaluated binary classification model performance with sensitivity, specificity, accuracy
1173 and f1-score metrics. Sensitivity, specificity, accuracy and f1-score were determined for every
1174 test run. The 95% confidence intervals of all performance metrics were estimated using the
1175 bootstrapping approach⁷⁴. For AML and ALL, 100 permutations per scenario were run for each
1176 scenario. For TB the performance metrics were collected by running 10 permutations for
1177 scenarios 1 to 4 and 5 permutations for scenarios 5 to 10. For COVID-19 the performance
1178 metrics were collected by running 20 permutations for each scenario. All metrics are listed in
1179 **Supplementary Tables 3 and 4**.

1180 Differences in performance metrics were tested using the Wilcoxon signed rank test with
1181 continuity correction (Individual Comparisons by Ranking Methods, Frank Wilcoxon,
1182 <https://sci2s.ugr.es/keel/pdf/algorithm/articulo/wilcoxon1945.pdf>). All test results are provided
1183 in **Supplementary Table 5**.

1184 To run the experiments, we used Python version 3.6.9 with Keras version 2.3.1 and
1185 Tensorflow version 2.2.0-rc2. We used scikit-learn library version 0.23.1⁷⁵ to calculate values
1186 for the metrics. Summary statistics and hypothesis tests were calculated using R version 3.5.2
1187 (R: A language and environment for statistical computing, <http://www.R-project.org/>, 2015).
1188 Calculation of each metric was done as follows:

1189

$$Sensitivity = \frac{TP}{TP + FN}$$

1190

$$Specificity = \frac{TN}{TN + FP}$$

1191

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

1192

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

1193

$$F1-score = \frac{2TP}{FP + FN + 2TP}$$

1194

where TP=True Positive, FP=False Positive, TN=True Negative, FN=False Negative

1195

1196 Data visualization

1197

The classification report and confusion matrix was generated with scikit-learn APIs for each permutation. Measurements of sensitivity, specificity and accuracy of each permutation run was read into a table in Excel using Power Query and used for visualization for the different scenarios in Power BI [Version: 2.81.5831.821 64-bit (Mai 2020)] with Box and Whisker chart by MAQ Software (<https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104381351>).

1203

1204

Data and software availability:

1205

Processed data can be accessed via the SuperSeries GSE122517 or via the individual SubSeries GSE122505 (dataset A1), GSE122511 (dataset A2) and GSE122515 (dataset A3).

1207

Dataset B consists of the following series which can be accessed at GEO: GSE101705, GSE107104, GSE112087, GSE128078, GSE66573, GSE79362, GSE84076, and GSE89403.

1209

Furthermore, it contains the Rhineland study. This dataset is not publicly available because of data protection regulations. Access to data can be provided to scientists in accordance with

1211

the Rhineland Study's Data Use and Access Policy. Requests for further information or to access the Rhineland Study's dataset should be directed to RS-DUAC@dzne.de. Dataset C

1213

contains dataset B and additional samples for COVID-19. These datasets are made available at the European Genome-Phenome Archive (EGA) under accession number

1215

EGAS00001004502, which is hosted by the EBI and the CRG.

1216 The code for preprocessing and for predictions can be found at GitHub
1217 (https://github.com/schultzelab/swarm_learning).

1218

1219

1220 **Supplementary Tables**

1221 Supplementary Table 1:	Overview over all sample numbers and scenarios
1222 Supplementary Table 2:	Dataset annotations of Dataset A, B and C
1223 Supplementary Table 3:	Prediction results for all scenarios and permutations
1224 Supplementary Table 4:	Summary statistics on all prediction scenarios
1225 Supplementary Table 5:	Statistical tests comparing single node vs. swarm predictions
1226 Supplementary Table 6:	Covid 19 Patient characteristics

1227 **References**

- 1228 1. Haendel, M. A., Chute, C. G. & Robinson, P. N. Classification, ontology, and precision
1229 medicine. *N. Engl. J. Med.* **379**, 1452–1462 (2018).
- 1230 2. Aronson, S. J. & Rehm, H. L. Building the foundation for genomics in precision
1231 medicine. *Nature* **526**, 336–342 (2015).
- 1232 3. Courtiol, P. *et al.* Deep learning-based classification of mesothelioma improves
1233 prediction of patient outcome. *Nat. Med.* **25**, 1519–1525 (2019).
- 1234 4. Liang, H. *et al.* Evaluation and accurate diagnoses of pediatric diseases using artificial
1235 intelligence. *Nat. Med.* **25**, 433–438 (2019).
- 1236 5. Wiens, J. *et al.* Do no harm: a roadmap for responsible machine learning for health
1237 care. *Nat. Med.* **25**, (2019).
- 1238 6. He, J. *et al.* The practical implementation of artificial intelligence technologies in
1239 medicine. *Nat. Med.* **25**, 30–36 (2019).
- 1240 7. Babic, B., Gerke, S., Evgeniou, T. & Glenn Cohen, I. Algorithms on regulatory lockdown
1241 in medicine. *Science* **366**, 1202–1204 (2019).
- 1242 8. Price, W. N. & Cohen, I. G. Privacy in the age of medical big data. *Nat. Med.* **25**, 37–43
1243 (2019).
- 1244 9. Zhang, X. *et al.* Viral and host factors related to the clinical outcome of COVID-19.
1245 *Nature* (2020) doi:10.1038/s41586-020-2355-0.
- 1246 10. Berlin, D. A., Gulick, R. M. & Martinez, F. J. Severe Covid-19. *N. Engl. J. Med.* (2020)
1247 doi:10.1056/nejmcp2009575.
- 1248 11. Gandhi, R. T., Lynch, J. B. & del Rio, C. Mild or Moderate Covid-19. *N. Engl. J. Med.*
1249 (2020) doi:10.1056/nejmcp2009249.
- 1250 12. Cho, A. AI systems aim to sniff out coronavirus outbreaks. *Science* **368**, 810–811
1251 (2020).
- 1252 13. Peiffer-Smadja, N. *et al.* Machine Learning for COVID-19 needs global collaboration
1253 and data-sharing. *Nat. Mach. Intell.* 1–2 (2020) doi:10.1038/s42256-020-0181-6.
- 1254 14. Luengo-Oroz, M. *et al.* Artificial intelligence cooperation to support the global response
1255 to COVID-19. *Nat. Mach. Intell.* 1–3 (2020) doi:10.1038/s42256-020-0184-3.
- 1256 15. Hu, Y. *et al.* The challenges of deploying artificial intelligence models in a rapidly
1257 evolving pandemic. *Nat. Mach. Intell.* **2**, 298–300 (2020).
- 1258 16. Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep
1259 learning. *Nature* **577**, 706–710 (2020).
- 1260 17. Ge, Y. *et al.* A data-driven drug repositioning framework discovered a potential
1261 therapeutic agent targeting COVID-19. *bioRxiv* (2020) doi:10.1101/2020.03.11.986836.
- 1262 18. Ting, D. S. W., Carin, L., Dzau, V. & Wong, T. Y. Digital technology and COVID-19. *Nat.*
1263 *Med.* **26**, 459–461 (2020).
- 1264 19. Cohen, I. G., Gostin, L. O. & Weitzner, D. J. Digital Smartphone Tracking for COVID-
1265 19: Public Health and Civil Liberties in Tension. *JAMA* (2020)
1266 doi:10.1001/jama.2020.8570.
- 1267 20. Jia, J. S. *et al.* Population flow drives spatio-temporal distribution of COVID-19 in China.
1268 *Nature* (2020) doi:10.1038/s41586-020-2284-y.
- 1269 21. Mei, X. *et al.* Artificial intelligence-enabled rapid diagnosis of patients with COVID-19.
1270 *Nat. Med.* (2020) doi:10.1038/s41591-020-0931-3.

- 1271 22. Zhang, K. *et al.* Clinically Applicable AI System for Accurate Diagnosis, Quantitative
1272 Measurements, and Prognosis of COVID-19 Pneumonia Using Computed
1273 Tomography. *Cell* **181**, (2020).
- 1274 23. Amisha, Malik, P., Pathania, M. & Rathaur, V. Overview of artificial intelligence in
1275 medicine. *J. Fam. Med. Prim. Care* **8**, 2328 (2019).
- 1276 24. Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H. & Aerts, H. J. W. L. Artificial
1277 intelligence in radiology. *Nature Reviews Cancer*. **18**, 500–510 (2018).
- 1278 25. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature*. **521**, 436–444 (2015).
- 1279 26. Kaassis, G. A., Makowski, M. R., Rückert, D. & Braren, R. F. Secure, privacy-preserving
1280 and federated machine learning in medical imaging. *Nat. Mach. Intell.* (2020)
1281 doi:10.1038/s42256-020-0186-1.
- 1282 27. Rajkomar, A., Dean, J. & Kohane, I. Machine learning in medicine. *New England
1283 Journal of Medicine* vol. 380 1347–1358 (2019).
- 1284 28. Obermeyer, Z. & Emanuel, E. J. Predicting the future-big data, machine learning, and
1285 clinical medicine. *New England Journal of Medicine*. **375**, 1216–1219 (2016).
- 1286 29. Savage, N. Machine learning: Calculating disease. *Nature*. **550**, S115–S117 (2017).
- 1287 30. Ping, P., Hermjakob, H., Polson, J. S., Benos, P. V. & Wang, W. Biomedical informatics
1288 on the cloud: A treasure hunt for advancing cardiovascular medicine. *Circ. Res.* **122**,
1289 1290–1301 (2018).
- 1290 31. Chouvarda, I. *et al.* WELCOME - Innovative integrated care platform using wearable
1291 sensing and smart cloud computing for COPD patients with Comorbidities. in *2014 36th
1292 Annual International Conference of the IEEE Engineering in Medicine and Biology
1293 Society, EMBC 2014* vol. 2014 3180–3183 (Institute of Electrical and Electronics
1294 Engineers Inc., 2014).
- 1295 32. Grossman, R. L. & White, K. P. A vision for a biomedical cloud. in *Journal of Internal
1296 Medicine*. **271**, 122–130 (J Intern Med, 2012).
- 1297 33. Char, D. S., Shah, N. H. & Magnus, D. Implementing machine learning in health care
1298 addressing ethical challenges. *New England Journal of Medicine*. **378**, 981–983 (2018).
- 1299 34. Gibney, E. The battle for ethical AI at the world's biggest machine-learning conference.
1300 *Nature*. **577**, 609 (2020).
- 1301 35. Finlayson, S. G. *et al.* Adversarial attacks on medical machine learning. *Science*. **363**,
1302 1287–1289 (2019).
- 1303 36. Ienca, M. & Vayena, E. On the responsible use of digital data to tackle the COVID-19
1304 pandemic. *Nat. Med.* **26**, 463–464 (2020).
- 1305 37. Konečný, J., McMahan, H. B., Ramage, D. & Richtárik, P. Federated Optimization:
1306 Distributed Machine Learning for On-Device Intelligence. (2016).
- 1307 38. Konečný, J. *et al.* Federated Learning: Strategies for Improving Communication
1308 Efficiency. (2016).
- 1309 39. McMahan, H. B., Moore, E., Ramage, D., Hampson, S. & Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proc.
1310 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017* (2016).
- 1311 40. Shokri, R. & Shmatikov, V. Privacy-Preserving Deep Learning | Proceedings of the 22nd
1312 ACM SIGSAC Conference on Computer and Communications Security. *ACM Digit.
1313 Libr.* 1310–1321 (2015).
- 1314 41. Pathak, M., Rane, S., Sun, W. & Raj, B. Privacy preserving probabilistic inference with
1315 Hidden Markov Models. in *ICASSP, IEEE International Conference on Acoustics,
1316*

- 1317 *Speech and Signal Processing - Proceedings* 5868–5871 (2011).
1318 doi:10.1109/ICASSP.2011.5947696.
- 1319 42. Chaussabel, D., Pascual, V. & Banchereau, J. Assessing the human immune system
1320 through blood transcriptomics. *BMC Biology*. **8**, (2010).
- 1321 43. Chaussabel, D. Assessment of immune status using blood transcriptomics and
1322 potential implications for global health. *Seminars in Immunology*. **27**, 58–66 (2015).
- 1323 44. Wei, Z. *et al.* Large sample size, wide variant spectrum, and advanced machine-
1324 learning technique boost risk prediction for inflammatory bowel disease. *Am. J. Hum. Genet.* **92**, 1008–1012 (2013).
- 1325 45. Stephens, Z. D. *et al.* Big data: Astronomical or genomic? *PLoS Biol.* **13**, (2015).
- 1326 46. Dove, E. S. *et al.* Genomic cloud computing: Legal and ethical points to consider. *Eur. J. Hum. Genet.* **23**, 1271–1278 (2015).
- 1327 47. Warnat-Herresthal, S. *et al.* Scalable Prediction of Acute Myeloid Leukemia Using High-
1328 Dimensional Machine Learning and Blood Transcriptomics. *iScience* **23**, (2020).
- 1329 48. Zak, D. E. *et al.* A blood RNA signature for tuberculosis disease risk: a prospective
1330 cohort study. *Lancet* **387**, 2312–2322 (2016).
- 1331 49. Leong, S. *et al.* Existing blood transcriptional classifiers accurately discriminate active
1332 tuberculosis from latent infection in individuals from south India. *Tuberculosis* **109**, 41–
1333 51 (2018).
- 1334 50. de Araujo, L. S. *et al.* Transcriptomic biomarkers for tuberculosis: Evaluation of DOCK9,
1335 EPHA4, and NPC2 mRNA expression in peripheral blood. *Front. Microbiol.* **7**, (2016).
- 1336 51. Verma, S. *et al.* ‘Tuberculosis in advanced HIV infection is associated with increased
1337 expression of IFNy and its downstream targets’. *BMC Infect. Dis.* **18**, (2018).
- 1338 52. Thompson, E. G. *et al.* Host blood RNA signatures predict the outcome of tuberculosis
1339 treatment. *Tuberculosis* **107**, 48–58 (2017).
- 1340 53. Corman, V. M. *et al.* Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-
1341 PCR. *Eurosurveillance* **25**, (2020).
- 1342 54. Schulte-Schrepping, J. *et al.* Suppressive myeloid cells are a hallmark of severe
1343 COVID-19. *medRxiv* 2020.06.03.20119818 (2020) doi:10.1101/2020.06.03.20119818.
- 1344 55. McKinney, S. M. *et al.* International evaluation of an AI system for breast cancer
1345 screening. *Nature* **577**, 89–94 (2020).
- 1346 56. Kerney, D. S. *et al.* Identifying Medical Diagnoses and Treatable Diseases by Image-
1347 Based Deep Learning. *Cell* **172**, 1122-1131.e9 (2018).
- 1348 57. Ardila, D. *et al.* End-to-end lung cancer screening with three-dimensional deep learning
1349 on low-dose chest computed tomography. *Nat. Med.* **25**, 954–961 (2019).
- 1350 58. Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural
1351 networks. *Nature* **542**, 115–118 (2017).
- 1352 59. Kaassis, G. *et al.* A machine learning model for the prediction of survival and tumor
1353 subtype in pancreatic ductal adenocarcinoma from preoperative diffusion-weighted
1354 imaging. *Eur. Radiol. Exp.* **3**, (2019).
- 1355 60. Kaassis, G. *et al.* A machine learning algorithm predicts molecular subtypes in
1356 pancreatic ductal adenocarcinoma with differential response to gemcitabine-based
1357 versus FOLFIRINOX chemotherapy. *PLoS One* **14**, (2019).
- 1358 61. Cui, E. *et al.* Predicting the ISUP grade of clear cell renal cell carcinoma with
1359 multiparametric MR and multiphase CT radiomics. *Eur. Radiol.* **30**, 2912–2921 (2020).
- 1360
- 1361

- 1362 62. Lu, H. *et al.* A mathematical-descriptor of tumor-mesoscopic-structure from computed-
1363 tomography images annotates prognostic- and molecular-phenotypes of epithelial
1364 ovarian cancer. *Nat. Commun.* **10**, (2019).
- 1365 63. Elshafeey, N. *et al.* Multicenter study demonstrates radiomic features derived from
1366 magnetic resonance perfusion images identify pseudoprogression in glioblastoma. *Nat.*
1367 *Commun.* **10**, (2019).
- 1368 64. Ryffel, T., Dufour-Sans, E., Gay, R., Bach, F. & Pointcheval, D. Partially Encrypted
1369 Machine Learning using Functional Encryption. (2019).
- 1370 65. Salem, M., Taheri, S. & Yuan, J.-S. Utilizing Transfer Learning and Homomorphic
1371 Encryption in a Privacy Preserving and Secure Biometric Recognition System.
1372 *Computers* **8**, 3 (2018).
- 1373 66. Dahl, M. *et al.* Private Machine Learning in TensorFlow using Secure Computation.
1374 (2018).
- 1375 67. Shi, W. *et al.* Deep Learning-Based Quantitative Computed Tomography Model in
1376 Predicting the Severity of COVID-19: A Retrospective Study in 196 Patients. *SSRN*
1377 *Electron. J.* (2020) doi:10.2139/ssrn.3546089.
- 1378 68. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene
1379 expression and hybridization array data repository. *Nucleic Acids Res.* (2002)
1380 doi:10.1093/nar/30.1.207.
- 1381 69. Irizarry R. A. Exploration, Normalization, and Summaries of High Density
1382 Oligonucleotide Array Probe Level Data. *Biostatistics* (2003).
- 1383 70. Gautier L. Affy--Analysis of Affymetrix GeneChip Data at the Probe Level.
1384 *Bioinformatik*s (2004).
- 1385 71. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion
1386 for RNA-seq data with DESeq2. *Genome Biol.* **15**, (2014).
- 1387 72. Dobin, A. *et al.* STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21
1388 (2013).
- 1389 73. Zwiener, I., Frisch, B. & Binder, H. Transforming RNA-Seq data to improve the
1390 performance of prognostic gene signatures. *PLoS One* **9**, (2014).
- 1391 74. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **7**, 1–26 (1979).
- 1392 75. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine*
1393 *Learning Research*. **12**, (2011).
- 1394