



Bionano Access API Guide

Document Number: 30462

Document Revision: A

Contents

Revision History	4
Introduction	4
Security	4
API Calls	4
<access>/administration/api/1.4/login	4
<access>/Bnx/api/2.0/getProjects	5
<access>/Bnx/api/2.0/createProject	6
<access>/Bnx/api/2.0/getSamples?projectuid={UID}	7
<access>/Bnx/api/2.0/createSample	8
<access>/Bnx/api/2.0/getMqrReport	9
<access>/Bnx/api/2.0/getFSHDRReport.....	15
<access>/Bnx/api/2.0/getFragileXReport	31
Technical Assistance	50

Legal Notice

For Research Use Only. Not for use in diagnostic procedures.

This material is protected by United States Copyright Law and International Treaties. Unauthorized use of this material is prohibited. No part of the publication may be copied, reproduced, distributed, translated, reverse-engineered or transmitted in any form or by any media, or by any means, whether now known or unknown, without the express prior permission in writing from Bionano Genomics. Copying, under the law, includes translating into another language or format. The technical data contained herein is intended for ultimate destinations permitted by U.S. law. Diversion contrary to U. S. law prohibited. This publication represents the latest information available at the time of release. Due to continuous efforts to improve the product, technical changes may occur that are not reflected in this document. Bionano Genomics reserves the right to make changes in specifications and other information contained in this publication at any time and without prior notice. Please contact Bionano Genomics Customer Support for the latest information.

BIONANO GENOMICS DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS DOCUMENT, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TO THE FULLEST EXTENT ALLOWED BY LAW, IN NO EVENT SHALL BIONANO GENOMICS BE LIABLE, WHETHER IN CONTRACT, TORT, WARRANTY, OR UNDER ANY STATUTE OR ON ANY OTHER BASIS FOR SPECIAL, INCIDENTAL, INDIRECT, PUNITIVE, MULTIPLE OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING FROM THIS DOCUMENT, INCLUDING BUT NOT LIMITED TO THE USE THEREOF, WHETHER OR NOT FORESEEABLE AND WHETHER OR NOT BIONANO GENOMICS IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Patents

Products of Bionano Genomics® may be covered by one or more U.S. or foreign patents.

Trademarks

The Bionano Genomics logo and names of Bionano Genomics products or services are registered trademarks or trademarks owned by Bionano Genomics in the United States and certain other countries.

Bionano Genomics®, Irys®, IrysView®, IrysChip®, IrysPrep®, IrysSolve®, Saphyr®, Saphyr Chip®, Bionano Access®, and Bionano EnFocus™ are trademarks of Bionano Genomics, Inc. All other trademarks are the sole property of their respective owners.

No license to use any trademarks of Bionano Genomics is given or implied. Users are not permitted to use these trademarks without the prior written consent of Bionano Genomics. The use of these trademarks or any other materials, except as permitted herein, is expressly prohibited and may be in violation of federal or other applicable laws.

© Copyright 2021 Bionano Genomics, Inc. All rights reserved.

Revision History

Revision	Release Date	Notes
A	10/20/2021	Initial release of document.

Introduction

The Bionano Access API Guide is being released in conjunction with Bionano Access 1.7. In this version of Access we have added several API calls to facilitate integration with most LIMS systems. This document will outline how to utilize the calls that have been added.

Security

All communication with the Bionano Access server is encrypted. The API calls are standard HTTPS request-based calls that receive and return data in JSON format. To use any API call you must first login to obtain a valid token which must be included in your subsequent requests. **IMPORTANT:** No protected health information should be injected into Bionano Access. All samples and project names should be de-identified appropriately.

API Calls

This section will inventory each API call and provide request and response examples. Remember that you must call the login API first then include the token in your subsequent calls. Replace '<access>' in the API URL with the address to your local Bionano Access instance. In the case of an error, the response will include a JSON that includes a StatusCode and Status values that provide the error code and message.

<access>/administration/api/1.4/login

Notes:

This GET request will return user and token information when valid credentials are provided. The token and user information should be included in subsequent requests as cookie values. Be aware the system will utilize request fingerprinting to prevent token hijacking.

Example Request:

```
{
  "username": "sway",
  "password": "Saphyr"
}
```

Example Response:

```
{
```



```
"created_at": "2021-05-27T22:16:19.634Z",
"updated_at": "2021-05-27T22:16:19.634Z",
"projectuid": "2992fd10-bf39-11eb-a7bb-3cfdfe7b4eb0"
},
{
  "name": "bionano_somatic",
  "description": "",
  "projectkey": "",
  "created_at": "2021-06-02T16:51:24.467Z",
  "updated_at": "2021-06-02T16:51:24.467Z",
  "projectuid": "c4088b02-c3c2-11eb-bd46-3cfdfe7b4eb0"
}
]
```

<access>/Bnx/api/2.0/createProject

Notes:

This POST request will return a JSON array of projects in the system.

Example Request:

```
{
  "name": "Test Project",
  "description": "This is a test project.",
  "projectkey": "test"
}
```

Example Response:

```
{
  "output": {
    "StatusCode": 0,
    "ProjectUID": "73603960-ee4d-11eb-831f-3cfdfe7b4eb0",
    "Status": "Project created."
  }
}
```

<access>/Bnx/api/2.0/getSamples?projectuid={UID}

Notes:

This GET request will return a JSON array of samples in the corresponding project. Include the project universal identifier on the URL as a query string parameter.

Example Response:

```
[
  {
    "sampleuid": "234d54ea-40b4-11eb-a0af-3cfdfe7b4eb0",
    "samplename": "Mus_musculus",
    "samplotype": "blood",
    "sampleindication": "",
    "sampledescription": "",
    "comments": "",
    "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",
    "created_at": "2020-12-17T22:06:39.546Z",
    "updated_at": "2020-12-17T22:06:39.546Z"
  },
  {
    "sampleuid": "234d5d5a-40b4-11eb-aaa3-3cfdfe7b4eb0",
    "samplename": "Mother",
    "samplotype": "blood",
    "sampleindication": "cancer",
    "sampledescription": "",
    "comments": "",
    "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",
    "created_at": "2020-12-17T22:06:39.548Z",
    "updated_at": "2020-12-17T22:06:39.548Z"
  }
]
```

<access>/Bnx/api/2.0/createSample

Notes:

This POST request will create a new sample. The sampledescription, sampletype, sampleindication, and comments fields are optional.

Example Request:

```
{
  "samplename": "ABX-3881-4FN",
  "sampledescription": "sample description goes here",
  "sampletype": "blood",
  "sampleindication": "cancer",
  "comments": "this is a comment",
  "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",
  "sampleuid": "d27ea170-ee52-11eb-9a03-0242ac130003"
}
```

Example Response:

```
{
  "output": {
    "StatusCode": 0,
    "SampleUID": "d27ea170-ee52-11eb-9a03-0242ac130003",
    "Status": "Sample created."
  }
}
```


Informatic Reports

Most Bionano Solve operations will generate an informatics report. These reports may include useful QC data you may want to extract. These reports can be retrieved in JSON format by providing the project universal identifier and or sample universal identifier. The system will return an array of reports. As of Bionano Access version 1.7 we have provided access to the reports for the Molecule Quality Report (MQR), EnFocus FSHD Analysis, and EnFocus Fragile X Analysis operations. The request is the same, but the response will vary based on the type of operation. We have provided an example for each report below.

`<access>/Bnx/api/2.0/getMqrReport`

Example Request:

```
{
  "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",
  "sampleuid": "529b9a2a-3fee-11eb-b9e6-3cfdfe7b4eb0"
}
```

Example Response (MQR):

```
[
  {
    "job": {
      "label": "Job Details",
      "value": {
        "jobpk": {
          "label": "Job ID",
          "value": 47,
          "description": "Job Identifier",
          "type": "integer"
        },
        "servername": {
          "label": "Server name",
          "value": "192.168.49.224",
          "description": "Name or IP of the server that ran job",
          "type": "string"
        },
        "created_at": {
          "label": "Created at",

```

```
"value": "2021-05-18T20:46:39.365Z",
"description": "Date job was created",
"type": "decimal"
},
"fullname": {
  "label": "User Name",
  "value": "John Smith",
  "description": "Full name of user who launched job",
  "type": "string"
},
"operationname": {
  "label": "Job type",
  "value": "Import Molecule",
  "description": "The type of operation performed",
  "type": "string"
},
"access": {
  "label": "Access Version",
  "value": "1.7",
  "description": "Bionano Access Version",
  "type": "string"
},
"solve": {
  "label": "Solve Version",
  "value": "Solve3.7_04072021",
  "description": "Bionano Solve Version",
  "type": "string"
},
"rescale": {
  "label": "Compute On Demand Version",
  "value": "Solve3.6_09252020",
  "description": "Version of pipeline for Compute On Demand",
  "type": "string"
},
"objectalias": {
  "label": "Job Name",
  "value": "asdfasdfs",
  "description": "Alias for Job",
```

```
    "type": "string"
  },
  "name": {
    "label": "Project Name",
    "value": "1.7",
    "description": "Name of the project",
    "type": "string"
  },
  "samplename": {
    "label": "Sample Name",
    "value": "one",
    "description": "Name of the sample",
    "type": "string"
  },
  "sampleuid": {
    "label": "Sample UID",
    "value": "bb1786c6-ad18-11eb-a4d3-3cfdfe7f3f60",
    "description": "System generated global unique identifier",
    "type": "string"
  },
  "reference": {
    "label": "Reference",
    "value": "hg38_DLE1_0kb_0labels.cmap",
    "description": "Name of the reference genome this sample was aligned to",
    "type": "string"
  }
},
"type": "section"
},
"mqr": {
  "label": "MQR Report Details",
  "value": {
    "ref_length": {
      "label": "Reference Length",
      "description": "Total length of reference sequence",
      "units": "bp",
      "orderPriority": 3,
      "value": 3088269832,
```

```
"type": "integer"
},
"mol_n_50_ge_20kb": {
  "label": "N50 (>= 20 kbp)",
  "description": "N50 of the molecules that are 20kbp or longer)",
  "units": "kbp",
  "orderPriority": 7,
  "value": 233.5504,
  "type": "decimal"
},
"quantity_ge_20kb": {
  "label": "Total DNA (>= 20kbp)",
  "description": "Total amount of DNA from molecules that are 20 kbp or longer",
  "units": "Gbp",
  "orderPriority": 8,
  "value": 268.84584102779996,
  "type": "decimal"
},
"mol_n_50": {
  "label": "N50 (>= 150kbp)",
  "description": "N50 of DNA molecules that are 150kbp or longer",
  "units": "kbp",
  "orderPriority": 9,
  "value": 235.8215,
  "type": "decimal"
},
"quantity": {
  "label": "Total DNA (>= 150kbp)",
  "description": "Total amount of DNA from molecules that are 150kbp or longer",
  "units": "Gbp",
  "orderPriority": 10,
  "value": 262.9716045002,
  "type": "decimal"
},
"mol_n_50_ge_150kb_site9": {
  "label": "N50 (>= 150kbp and min sites >=9)",
  "description": "Same as other N50 fields, but molecules must have at least 9 labels",
  "units": "kbp",
```

```
"orderPriority": 11,
"value": 235.8215,
"type": "decimal"
},
"quantity_ge_150kb_site9": {
  "label": "Total DNA (>= 150kbp and min sites >= 9)",
  "description": "Same as other Total DNA fields, but molecules must have at least 9 labels",
  "units": "Gbp",
  "orderPriority": 12,
  "value": 262.9716045002,
  "type": "decimal"
},
"map_rate": {
  "label": "Map rate",
  "description": "Percentage of molecules that are 150kbp or longer mapped to the reference",
  "units": "%",
  "orderPriority": 13,
  "value": 91.2,
  "type": "decimal"
},
"avg_label_density": {
  "label": "Average label density (>= 150kbp)",
  "description": "Average number of labels per 100 kbp for the molecules that are 150kbp or longer",
  "units": "/100kbp",
  "orderPriority": 15,
  "value": 17.39,
  "type": "decimal"
},
"bpp": {
  "label": "Base pairs per pixel",
  "description": "Calculated base pairs per pixel in the alignment by comparing molecule",
  "value": 512.83,
  "type": "decimal"
},
"scaling_sd": {
  "label": "Scaling SD",
  "description": "Linear term in sizing error relative to reference",
  "value": 0.02286,
```

```
    "type": "decimal"
  },
  "relative_sd": {
    "label": "Relative SD",
    "description": "Quadratic term in sizing error relative to reference",
    "value": 0.03471,
    "type": "decimal"
  },
  "sf": {
    "label": "Site SD",
    "description": "Constant term in sizing error relative to reference",
    "value": 0.14211,
    "type": "decimal"
  },
  "sr": {
    "label": "Relative SD",
    "description": "Quadratic term in sizing error relative to reference",
    "value": 0.03471,
    "type": "decimal"
  },
  "fp_rate": {
    "label": "Positive label variance (PLV)",
    "description": "Percentage of labels absent in reference",
    "value": 6.597,
    "type": "decimal"
  },
  "fn_rate": {
    "label": "Negative label variance (NLV)",
    "description": "Percentage of reference labels absent in molecules",
    "value": 6.647,
    "type": "decimal"
  }
}
"type": "section"
}
}
```

<access>/Bnx/api/2.0/getFSHDRReport

Example Request:

```
{  
  "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",  
  "sampleuid": "529b9a2a-3fee-11eb-b9e6-3cfdfe7b4eb0"  
}
```

Example Response (FSHD):

```
[  
  {  
    "job": {  
      "label": "Job Details",  
      "value": {  
        "jobpk": {  
          "label": "Job ID",  
          "value": 39,  
          "description": "Job Identifier",  
          "type": "integer"  
        },  
        "created_at": {  
          "label": "Created at",  
          "value": "2021-02-02T14:59:18.907Z",  
          "description": "Date job was created",  
          "type": "decimal"  
        },  
        "fullname": {  
          "label": "User Name",  
          "value": "Bionano Development",  
          "description": "Full name of user who launched job",  
          "type": "string"  
        },  
        "operationname": {  
          "label": "Job type",  
          "value": "Import EnFocus™ FSHD Analysis",  
        }  
      }  
    }  
  }  
]
```

```
"description": "The type of operation performed",
"type": "string"
},
"access": {
  "label": "Access Version",
  "value": "1.7",
  "description": "Bionano Access Version",
  "type": "string"
},
"solve": {
  "label": "Solve Version",
  "value": "Solve3.6.1",
  "description": "Bionano Solve Version",
  "type": "string"
},
"rescale": {
  "label": "Compute On Demand Version",
  "value": "Solve3.6_09252020",
  "description": "Version of pipeline for Compute On Demand",
  "type": "string"
},
"objectalias": {
  "label": "Job Name",
  "value": "fshd 3",
  "description": "Alias for Job",
  "type": "string"
},
"name": {
  "label": "Project Name",
  "value": "1.7",
  "description": "Name of the project",
  "type": "string"
},
"samplename": {
  "label": "Sample Name",
  "value": "human",
  "description": "Name of the sample",
  "type": "string"
}
```



```
    },
    "sampleuid": {
      "label": "Sample UID",
      "value": "7b9a6850-4f9f-11eb-8358-3cfdfe7f3f60",
      "description": "System generated global unique identifier",
      "type": "string"
    },
  },
  "reference": {
    "label": "Reference",
    "value": "hg38_DLE1_0kb_0labels.cmap",
    "description": "Name of the reference genome this sample was aligned to",
    "type": "string"
  }
},
"type": "section"
},
"Experiment information": {
  "label": "Experiment information",
  "value": {
    "Sample name": {
      "label": "Sample name",
      "value": "Human",
      "type": "string",
      "description": "Name of the sample"
    },
    "Enzyme used": {
      "label": "Enzyme used",
      "value": "DLE-1",
      "type": "string",
      "description": "Enzyme used to label the DNA"
    },
    "Instrument serial number": {
      "label": "Instrument serial number",
      "value": "SAPHYR_D-BETA3",
      "type": "string",
      "description": "Serial number of the Bionano Saphyr instrument"
    },
    "Chip ID": {
```

```
"label": "Chip ID",
"value": "WARMA5ONPOYXZNWU (Flowcell 3)",
"type": "string",
"description": "Serial number of the chip followed by the flowcell number in parentheses"
},
"Run ID": {
  "label": "Run ID",
  "value": "b7005743-2333-4e3b-83f6-0b47014bc4e1",
  "type": "string",
  "description": "Unique identifier for a chip run"
},
"Date of data collection": {
  "label": "Date of data collection",
  "value": "2020-07-10 05:11:17 PM",
  "type": "string",
  "description": "Date and time when the data from the first scan is generated"
},
"Version of ICS software": {
  "label": "Version of ICS software",
  "value": "ICS 4.8.19085.2",
  "type": "string",
  "description": "Instrument Control Software version used for analyzing the image data"
}
},
"type": "section"
},
"Overall sample quality metrics": {
  "label": "Overall sample quality metrics",
  "value": {
    "Inferred sex of sample": {
      "label": "Inferred sex of sample",
      "value": "female",
      "type": "string",
      "description": "Sex of the sample as inferred from the copy number analysis pipeline based on the molecule alignment ('coverage') data"
    },
    "Assessment of molecule quality": {
      "label": "Assessment of molecule quality",
```

```
    "value": "FAIL",
    "type": "string",
    "description": "Quality of the molecules based on three criteria: molecule N50 (> 150kbp) has to be at
least 200 kbp, effective coverage has to be at least 87.5X, and map rate has to be at least 70%"
  },
  "Assessment of stable regions": {
    "label": "Assessment of stable regions",
    "value": "PASS",
    "type": "string",
    "description": "Quality of the consensus based on evaluation of regions considered stable"
  }
},
"type": "section"
},
"Analysis information": {
  "label": "Analysis information",
  "value": {
    "Analysis performed": {
      "label": "Analysis performed",
      "value": "Bionano EnFocus™ FSHD Analysis",
      "type": "string",
      "description": "The type of operation performed"
    },
    "Job ID": {
      "label": "Job ID",
      "value": 4009,
      "type": "integer",
      "description": "Job unique identifier"
    },
    "Job name": {
      "label": "Job name",
      "value": "Cohort_FSHD_07132020",
      "type": "string",
      "description": "Alias for the job"
    },
    "Operator name": {
      "label": "Operator name",
      "value": "Beverlie Poblete",
```

```
    "type": "string",
    "description": "Full name of user who launched job"
  },
  "Date of analysis": {
    "label": "Date of analysis",
    "value": "2020-07-14 01:55",
    "type": "string",
    "description": "The date and time the job was started"
  },
  "Version of assembly pipeline": {
    "label": "Version of assembly pipeline",
    "value": "N/A",
    "type": "string",
    "description": "Version of the assembly pipeline used for targeted assembly of the regions of
interested"
  },
  "Version of FSHD analysis pipeline": {
    "label": "Version of FSHD analysis pipeline",
    "value": "Bionano EnFocus™ FSHD Analysis 1.0",
    "type": "string",
    "description": "Version of the analysis processing pipeline"
  }
},
"type": "section"
},
"Detailed results": {
  "label": "Detailed results",
  "value": [
    {
      "MapID": {
        "label": "MapID",
        "value": 21,
        "type": "integer"
      },
      "Chr": {
        "label": "Chr",
        "value": 4,
        "type": "integer"
      }
    }
  ]
}
```

```
},
"Haplotype": {
  "label": "Haplotype",
  "value": "4qA",
  "type": "string"
},
"Count_repeat": {
  "label": "Count_repeat",
  "value": 27,
  "type": "integer"
},
"Repeat_spanning_coverage": {
  "label": "Repeat_spanning_coverage",
  "value": 13,
  "type": "integer"
},
"Start_repeat": {
  "label": "Start_repeat",
  "value": 475,
  "type": "integer"
},
"End_repeat": {
  "label": "End_repeat",
  "value": 476,
  "type": "integer"
},
"Start_haplotype": {
  "label": "Start_haplotype",
  "value": 3093839.1,
  "type": "decimal"
},
"End_haplotype": {
  "label": "End_haplotype",
  "value": 3112150.5,
  "type": "decimal"
},
"Confidence": {
  "label": "Confidence",
```

```
"value": -1,
"type": "integer"
},
"Map_alignment_confidence": {
  "label": "Map_alignment_confidence",
  "value": 497.61,
  "type": "decimal"
},
"Anchor_to_mapend_map": {
  "label": "Anchor_to_mapend_map",
  "value": 106460.70000000019,
  "type": "decimal"
},
"Array_length": {
  "label": "Array_length",
  "value": 90.07,
  "type": "decimal"
},
"Count_length_consistency": {
  "label": "Count_length_consistency",
  "value": 1.01,
  "type": "decimal"
},
"Contains_SV": {
  "label": "Contains_SV",
  "value": true,
  "type": "decimal"
},
"ImageText": {
  "label": "ImageText",
  "value": "Chromosome 4, Map 21 whose haplotype is 4qA has a calculated repeat count of 27",
  "type": "string"
},
"Count_repeat_mol": {
  "label": "Count_repeat_mol",
  "value": -1,
  "type": "integer"
},
```

```
"Merged": {
  "label": "Merged",
  "value": false,
  "type": "decimal"
},
"truncated_bool": {
  "label": "truncated_bool",
  "value": false,
  "type": "decimal"
},
"parsed_repeat_counts": {
  "label": "parsed_repeat_counts",
  "value": -1,
  "type": "integer"
}
},
{
  "MapID": {
    "label": "MapID",
    "value": 22,
    "type": "integer"
  },
  "Chr": {
    "label": "Chr",
    "value": 4,
    "type": "integer"
  },
  "Haplotype": {
    "label": "Haplotype",
    "value": "4qA",
    "type": "string"
  },
  "Count_repeat": {
    "label": "Count_repeat",
    "value": 33,
    "type": "integer"
  },
  "Repeat_spanning_coverage": {
```

```
"label": "Repeat_spanning_coverage",
"value": 6,
"type": "integer"
},
"Start_repeat": {
  "label": "Start_repeat",
  "value": 396,
  "type": "integer"
},
"End_repeat": {
  "label": "End_repeat",
  "value": 397,
  "type": "integer"
},
"Start_haplotype": {
  "label": "Start_haplotype",
  "value": 2717339.4,
  "type": "decimal"
},
"End_haplotype": {
  "label": "End_haplotype",
  "value": 2735547.4,
  "type": "decimal"
},
"Confidence": {
  "label": "Confidence",
  "value": -1,
  "type": "integer"
},
"Map_alignment_confidence": {
  "label": "Map_alignment_confidence",
  "value": 410.01,
  "type": "decimal"
},
"Anchor_to_mapend_map": {
  "label": "Anchor_to_mapend_map",
  "value": 124772.1000000001,
  "type": "decimal"
}
```



```
  },
  "Array_length": {
    "label": "Array_length",
    "value": 108.38,
    "type": "decimal"
  },
  "Count_length_consistency": {
    "label": "Count_length_consistency",
    "value": 1,
    "type": "integer"
  },
  "Contains_SV": {
    "label": "Contains_SV",
    "value": true,
    "type": "decimal"
  },
  "ImageText": {
    "label": "ImageText",
    "value": "Chromosome 4, Map 22 whose haplotype is 4qA has a calculated repeat count of 33",
    "type": "string"
  },
  "Count_repeat_mol": {
    "label": "Count_repeat_mol",
    "value": -1,
    "type": "integer"
  },
  "Merged": {
    "label": "Merged",
    "value": false,
    "type": "decimal"
  },
  "truncated_bool": {
    "label": "truncated_bool",
    "value": false,
    "type": "decimal"
  },
  "parsed_repeat_counts": {
    "label": "parsed_repeat_counts",
```

```
    "value": -1,  
    "type": "integer"  
  }  
},  
{  
  "MapID": {  
    "label": "MapID",  
    "value": 11,  
    "type": "integer"  
  },  
  "Chr": {  
    "label": "Chr",  
    "value": 10,  
    "type": "integer"  
  },  
  "Haplotype": {  
    "label": "Haplotype",  
    "value": "10qA",  
    "type": "string"  
  },  
  "Count_repeat": {  
    "label": "Count_repeat",  
    "value": 16,  
    "type": "integer"  
  },  
  "Repeat_spanning_coverage": {  
    "label": "Repeat_spanning_coverage",  
    "value": 10,  
    "type": "integer"  
  },  
  "Start_repeat": {  
    "label": "Start_repeat",  
    "value": 173,  
    "type": "integer"  
  },  
  "End_repeat": {  
    "label": "End_repeat",  
    "value": 174,
```

```
"type": "integer"  
},  
"Start_haplotype": {  
  "label": "Start_haplotype",  
  "value": 1955435.2,  
  "type": "decimal"  
},  
"End_haplotype": {  
  "label": "End_haplotype",  
  "value": 1973966.1,  
  "type": "decimal"  
},  
"Confidence": {  
  "label": "Confidence",  
  "value": -1,  
  "type": "integer"  
},  
"Map_alignment_confidence": {  
  "label": "Map_alignment_confidence",  
  "value": 219.55,  
  "type": "decimal"  
},  
"Anchor_to_mapend_map": {  
  "label": "Anchor_to_mapend_map",  
  "value": 67916.5,  
  "type": "decimal"  
},  
"Array_length": {  
  "label": "Array_length",  
  "value": 52.08,  
  "type": "decimal"  
},  
"Count_length_consistency": {  
  "label": "Count_length_consistency",  
  "value": 0.99,  
  "type": "decimal"  
},  
"Contains_SV": {
```

```
    "label": "Contains_SV",
    "value": true,
    "type": "decimal"
  },
  "ImageText": {
    "label": "ImageText",
    "value": "Chromosome 10, Map 11 whose haplotype is 10qA has a calculated repeat count of 16",
    "type": "string"
  },
  "Count_repeat_mol": {
    "label": "Count_repeat_mol",
    "value": -1,
    "type": "integer"
  },
  "Merged": {
    "label": "Merged",
    "value": false,
    "type": "decimal"
  },
  "truncated_bool": {
    "label": "truncated_bool",
    "value": false,
    "type": "decimal"
  },
  "parsed_repeat_counts": {
    "label": "parsed_repeat_counts",
    "value": -1,
    "type": "integer"
  }
},
{
  "MapID": {
    "label": "MapID",
    "value": 12,
    "type": "integer"
  },
  "Chr": {
    "label": "Chr",
```

```
"value": 10,
  "type": "integer"
},
"Haplotype": {
  "label": "Haplotype",
  "value": "10qA",
  "type": "string"
},
"Count_repeat": {
  "label": "Count_repeat",
  "value": 21,
  "type": "integer"
},
"Repeat_spanning_coverage": {
  "label": "Repeat_spanning_coverage",
  "value": 12,
  "type": "integer"
},
"Start_repeat": {
  "label": "Start_repeat",
  "value": 406,
  "type": "integer"
},
"End_repeat": {
  "label": "End_repeat",
  "value": 407,
  "type": "integer"
},
"Start_haplotype": {
  "label": "Start_haplotype",
  "value": 3790797.2,
  "type": "decimal"
},
"End_haplotype": {
  "label": "End_haplotype",
  "value": 3809279.8,
  "type": "decimal"
},
```

```
"Confidence": {
  "label": "Confidence",
  "value": -1,
  "type": "integer"
},
"Map_alignment_confidence": {
  "label": "Map_alignment_confidence",
  "value": 517.32,
  "type": "decimal"
},
"Anchor_to_mapend_map": {
  "label": "Anchor_to_mapend_map",
  "value": 86447.40000000037,
  "type": "decimal"
},
"Array_length": {
  "label": "Array_length",
  "value": 70.61,
  "type": "decimal"
},
"Count_length_consistency": {
  "label": "Count_length_consistency",
  "value": 1.02,
  "type": "decimal"
},
"Contains_SV": {
  "label": "Contains_SV",
  "value": true,
  "type": "decimal"
},
"ImageText": {
  "label": "ImageText",
  "value": "Chromosome 10, Map 12 whose haplotype is 10qA has a calculated repeat count of 21",
  "type": "string"
},
"Count_repeat_mol": {
  "label": "Count_repeat_mol",
  "value": -1,
```

```
        "type": "integer"
      },
      "Merged": {
        "label": "Merged",
        "value": false,
        "type": "decimal"
      },
      "truncated_bool": {
        "label": "truncated_bool",
        "value": false,
        "type": "decimal"
      },
      "parsed_repeat_counts": {
        "label": "parsed_repeat_counts",
        "value": -1,
        "type": "integer"
      }
    }
  ],
  "type": "section"
}
]
```

<access>/Bnx/api/2.0/getFragileXReport

Example Request:

```
{
  "projectuid": "3b5c7000-3fee-11eb-b9e6-3cfdfe7b4eb0",
  "sampleuid": "529b9a2a-3fee-11eb-b9e6-3cfdfe7b4eb0"
}
```

Example Response (Fragile X):

```
[
  {
    "job": {
      "label": "Job Details",
```

```
"value": {
  "jobpk": {
    "label": "Job ID",
    "value": 554,
    "description": "Job Identifier",
    "type": "integer"
  },
  "servername": {
    "label": "Server name",
    "value": "192.168.49.224",
    "description": "Name or IP of the server that ran job",
    "type": "string"
  },
  "created_at": {
    "label": "Created at",
    "value": "2021-09-24T03:59:45.339Z",
    "description": "Date job was created",
    "type": "decimal"
  },
  "fullname": {
    "label": "User Name",
    "value": "dipa roychoudhury",
    "description": "Full name of user who launched job",
    "type": "string"
  },
  "operationname": {
    "label": "Job type",
    "value": "EnFocus™ Fragile X Analysis",
    "description": "The type of operation performed",
    "type": "string"
  },
  "access": {
    "label": "Access Version",
    "value": "1.7",
    "description": "Bionano Access Version",
    "type": "string"
  },
  "solve": {
```



```
"label": "Solve Version",
"value": "Solve3.7_09172021_175",
"description": "Bionano Solve Version",
"type": "string"
},
"rescale": {
  "label": "Compute On Demand Version",
  "value": "Solve3.7_09012021",
  "description": "Version of pipeline for Compute On Demand",
  "type": "string"
},
"objectalias": {
  "label": "Job Name",
  "value": "AML_632 - Fragile X Analysis",
  "description": "Alias for Job",
  "type": "string"
},
"name": {
  "label": "Project Name",
  "value": "alpha",
  "description": "Name of the project",
  "type": "string"
},
"samplename": {
  "label": "Sample Name",
  "value": "AML_632",
  "description": "Name of the sample",
  "type": "string"
},
"sampleuid": {
  "label": "Sample UID",
  "value": "53488f2c-d838-11eb-9c56-3cfdfe7f3f60",
  "description": "System generated global unique identifier",
  "type": "string"
},
"reference": {
  "label": "Reference",
  "value": "hg38_DLE1_0kb_0labels.cmap",
```

```
"description": "Name of the reference genome this sample was aligned to",
  "type": "string"
},
"inputjob1_jobpk": {
  "label": "Input job 1 - Job ID",
  "value": 539,
  "description": "Job Identifier",
  "type": "integer"
},
"inputjob1_objectalias": {
  "label": "Input job 1 - Job Name",
  "value": "AML_632 - Molecule Filter",
  "description": "Alias for Job",
  "type": "string"
},
"inputjob1_operationname": {
  "label": "Input job 1 - Job type",
  "value": "Filter BNX",
  "description": "The type of operation performed",
  "type": "string"
}
},
"type": "section"
},
"Experiment information": {
  "value": {
    "Sample name": {
      "label": "Sample name",
      "value": "AML_632",
      "type": "string",
      "description": "Name of the sample"
    },
    "Enzyme used": {
      "label": "Enzyme used",
      "value": "DLE1",
      "type": "string",
      "description": "Enzyme used to label the DNA"
    }
  }
},
```

```
"Instrument serial number": {
  "label": "Instrument serial number",
  "value": "SAPHYR_F06",
  "type": "string",
  "description": "Serial number of the Bionano Saphyr instrument"
},
"Chip ID": {
  "label": "Chip ID",
  "value": "UWME6LGNPMGUVNWU (Flowcell 1)",
  "type": "string",
  "description": "Serial number of the chip followed by the flowcell number in parentheses"
},
"Run ID": {
  "label": "Run ID",
  "value": "ff96e3ca-dcee-4f06-9cfe-921f084bc32f",
  "type": "string",
  "description": "Unique identifier for a chip run"
},
>Date of data collection": {
  "label": "Date of data collection",
  "value": "2020-02-01 07:11:15 AM",
  "type": "string",
  "description": "Date and time when the data from the first scan is generated"
},
"Version of ICS software": {
  "label": "Version of ICS software",
  "value": "ICS 4.9.19238.1",
  "type": "string",
  "description": "Instrument Control Software version used for analyzing the image data"
}
},
"type": "section",
"label": "Experiment information"
},
"Overall sample quality metrics": {
  "value": {
    "Inferred sex of sample": {
      "label": "Inferred sex of sample",
```

```
    "value": "female",
    "type": "string",
    "description": "Sex of the sample as inferred from the copy number analysis pipeline based on the
molecule alignment ('coverage') data"
  },
  "Assessment of molecule quality": {
    "label": "Assessment of molecule quality",
    "value": "PASS",
    "type": "string",
    "description": "Quality of the molecules based on three criteria: molecule N50 (> 150kbp) has to be at
least 200 kbp, effective coverage has to be at least 87.5X, and map rate has to be at least 70%"
  },
  "Assessment of stable regions": {
    "label": "Assessment of stable regions",
    "value": "PASS",
    "type": "string",
    "description": "Quality of the consensus based on evaluation of regions considered stable"
  }
},
"type": "section",
"label": "Overall sample quality metrics"
},
"Analysis information": {
  "value": {
    "Analysis performed": {
      "label": "Analysis performed",
      "value": "Bionano EnFocus™ Fragile X Analysis",
      "type": "string",
      "description": "The type of operation performed"
    },
    "Job ID": {
      "label": "Job ID",
      "value": 554,
      "type": "integer",
      "description": "Job unique identifier"
    },
    "Job name": {
      "label": "Job name",
```

```
"value": "AML_632 - Fragile X Analysis",
"type": "string",
"description": "Alias for the job"
},
"Operator name": {
  "label": "Operator name",
  "value": "dipa roychoudhury",
  "type": "string",
  "description": "Full name of user who launched job"
},
"Date of analysis": {
  "label": "Date of analysis",
  "value": "2021-09-24 10:34",
  "type": "string",
  "description": "The date and time the job was started"
},
"Version of Bionano Access": {
  "label": "Version of Bionano Access",
  "value": "1.7",
  "type": "string"
},
"Version of Bionano Solve": {
  "label": "Version of Bionano Solve",
  "value": "Solve3.7_09172021_175",
  "type": "string"
}
},
"type": "section",
"label": "Analysis information"
},
"Detailed results": {
  "value": [
    {
      "Gene": {
        "label": "Gene",
        "value": "FMR1",
        "type": "string"
      }
    }
  ]
},
```

```
"Sample": {
  "label": "Sample",
  "value": "AML_632",
  "type": "string"
},
"Sex": {
  "label": "Sex",
  "value": "female",
  "type": "string"
},
"Chr": {
  "label": "Chr",
  "value": "X",
  "type": "string"
},
"Start_ref": {
  "label": "Start_ref",
  "value": 147910189,
  "type": "integer"
},
"End_ref": {
  "label": "End_ref",
  "value": 147918814,
  "type": "integer"
},
"Interval_ref": {
  "label": "Interval_ref",
  "value": 8625,
  "type": "integer"
},
"Count_repeat_ref": {
  "label": "Count_repeat_ref",
  "value": 25,
  "type": "integer"
},
"Repeat_unit_size": {
  "label": "Repeat_unit_size",
  "value": 3,
```

```
    "type": "integer"
  },
  "Irrelevant_ref": {
    "label": "Irrelevant_ref",
    "value": 8550,
    "type": "integer"
  },
  "MapID": {
    "label": "MapID",
    "value": 231,
    "type": "integer"
  },
  "Start_repeat": {
    "label": "Start_repeat",
    "value": 299,
    "type": "integer"
  },
  "End_repeat": {
    "label": "End_repeat",
    "value": 300,
    "type": "integer"
  },
  "I_2_start_ref": {
    "label": "I_2_start_ref",
    "value": -1,
    "type": "integer"
  },
  "I_2_end_ref": {
    "label": "I_2_end_ref",
    "value": -1,
    "type": "integer"
  },
  "Array_length": {
    "label": "Array_length",
    "value": 8.95,
    "type": "decimal"
  },
  "Unmatched_labels": {
```

```
"label": "Unmatched_labels",
"value": 0,
"type": "integer"
},
"Count_repeat_observed": {
  "label": "Count_repeat_observed",
  "value": 133,
  "type": "integer"
},
"Count_repeat": {
  "label": "Count_repeat",
  "value": 133,
  "type": "integer"
},
"P >= expansion_cutoff": {
  "label": "P >= expansion_cutoff",
  "value": "1.57%",
  "type": "string"
},
"Expanded_repeat": {
  "label": "Expanded_repeat",
  "value": 200,
  "type": "integer"
},
"Realigned": {
  "label": "Realigned",
  "value": false,
  "type": "decimal"
},
"CI_lower": {
  "label": "CI_lower",
  "value": 59,
  "type": "integer"
},
"CI_upper": {
  "label": "CI_upper",
  "value": 211,
  "type": "integer"
}
```



```
    },
    "Percentile": {
      "label": "Percentile",
      "value": 100,
      "type": "integer"
    },
  },
  "Repeat_spanning_coverage": {
    "label": "Repeat_spanning_coverage",
    "value": 57,
    "type": "integer"
  },
  },
  "Qry_contig_length": {
    "label": "Qry_contig_length",
    "value": 2638189.3,
    "type": "decimal"
  },
  },
  "ImageText": {
    "label": "ImageText",
    "value": "Chromosome X, Map231 has a calculated repeat count of 133",
    "type": "string"
  }
},
{
  "Gene": {
    "label": "Gene",
    "value": "FMR1",
    "type": "string"
  },
  "Sample": {
    "label": "Sample",
    "value": "AML_632",
    "type": "string"
  },
  "Sex": {
    "label": "Sex",
    "value": "female",
    "type": "string"
  },
}
```

```
"Chr": {
  "label": "Chr",
  "value": "X",
  "type": "string"
},
"Start_ref": {
  "label": "Start_ref",
  "value": 147910189,
  "type": "integer"
},
"End_ref": {
  "label": "End_ref",
  "value": 147918814,
  "type": "integer"
},
"Interval_ref": {
  "label": "Interval_ref",
  "value": 8625,
  "type": "integer"
},
"Count_repeat_ref": {
  "label": "Count_repeat_ref",
  "value": 25,
  "type": "integer"
},
"Repeat_unit_size": {
  "label": "Repeat_unit_size",
  "value": 3,
  "type": "integer"
},
"Irrelevant_ref": {
  "label": "Irrelevant_ref",
  "value": 8550,
  "type": "integer"
},
"MapID": {
  "label": "MapID",
  "value": 232,
```

```
    "type": "integer"
  },
  "Start_repeat": {
    "label": "Start_repeat",
    "value": 267,
    "type": "integer"
  },
  "End_repeat": {
    "label": "End_repeat",
    "value": 268,
    "type": "integer"
  },
  "l_2_start_ref": {
    "label": "l_2_start_ref",
    "value": -1,
    "type": "integer"
  },
  "l_2_end_ref": {
    "label": "l_2_end_ref",
    "value": -1,
    "type": "integer"
  },
  "Array_length": {
    "label": "Array_length",
    "value": 8.95,
    "type": "decimal"
  },
  "Unmatched_labels": {
    "label": "Unmatched_labels",
    "value": 0,
    "type": "integer"
  },
  "Count_repeat_observed": {
    "label": "Count_repeat_observed",
    "value": 133,
    "type": "integer"
  },
  "Count_repeat": {
```

```
"label": "Count_repeat",
"value": 133,
"type": "integer"
},
"P >= expansion_cutoff": {
"label": "P >= expansion_cutoff",
"value": "1.56%",
"type": "string"
},
"Expanded_repeat": {
"label": "Expanded_repeat",
"value": 200,
"type": "integer"
},
"Realigned": {
"label": "Realigned",
"value": false,
"type": "decimal"
},
"CI_lower": {
"label": "CI_lower",
"value": 59,
"type": "integer"
},
"CI_upper": {
"label": "CI_upper",
"value": 211,
"type": "integer"
},
"Percentile": {
"label": "Percentile",
"value": 100,
"type": "integer"
},
"Repeat_spanning_coverage": {
"label": "Repeat_spanning_coverage",
"value": 30,
"type": "integer"
}
```

```
    },
    "Qry_contig_length": {
      "label": "Qry_contig_length",
      "value": 2678514.7,
      "type": "decimal"
    },
    "ImageText": {
      "label": "ImageText",
      "value": "Chromosome X, Map232 has a calculated repeat count of 133",
      "type": "string"
    }
  },
  {
    "Gene": {
      "label": "Gene",
      "value": "FMR1",
      "type": "string"
    },
    "Sample": {
      "label": "Sample",
      "value": "AML_632",
      "type": "string"
    },
    "Sex": {
      "label": "Sex",
      "value": "female",
      "type": "string"
    },
    "Chr": {
      "label": "Chr",
      "value": "X",
      "type": "string"
    },
    "Start_ref": {
      "label": "Start_ref",
      "value": 147910189,
      "type": "integer"
    }
  },
}
```

```
"End_ref": {
  "label": "End_ref",
  "value": 147918814,
  "type": "integer"
},
"Interval_ref": {
  "label": "Interval_ref",
  "value": 8625,
  "type": "integer"
},
"Count_repeat_ref": {
  "label": "Count_repeat_ref",
  "value": 25,
  "type": "integer"
},
"Repeat_unit_size": {
  "label": "Repeat_unit_size",
  "value": 3,
  "type": "integer"
},
"Irrelevant_ref": {
  "label": "Irrelevant_ref",
  "value": 9052,
  "type": "integer"
},
"MapID": {
  "label": "MapID",
  "value": 550,
  "type": "integer"
},
"Start_repeat": {
  "label": "Start_repeat",
  "value": 19,
  "type": "integer"
},
"End_repeat": {
  "label": "End_repeat",
  "value": 20,
```

```
"type": "integer"
},
"l_2_start_ref": {
  "label": "l_2_start_ref",
  "value": -1,
  "type": "integer"
},
"l_2_end_ref": {
  "label": "l_2_end_ref",
  "value": 502,
  "type": "integer"
},
"Array_length": {
  "label": "Array_length",
  "value": 9.06,
  "type": "decimal"
},
"Unmatched_labels": {
  "label": "Unmatched_labels",
  "value": 0,
  "type": "integer"
},
"Count_repeat_observed": {
  "label": "Count_repeat_observed",
  "value": 1,
  "type": "integer"
},
"Count_repeat": {
  "label": "Count_repeat",
  "value": 2,
  "type": "integer"
},
"P >= expansion_cutoff": {
  "label": "P >= expansion_cutoff",
  "value": "< 0.01%",
  "type": "string"
},
"Expanded_repeat": {
```

```
"label": "Expanded_repeat",
"value": 200,
"type": "integer"
},
"Realigned": {
  "label": "Realigned",
  "value": false,
  "type": "decimal"
},
"CI_lower": {
  "label": "CI_lower",
  "value": 0,
  "type": "integer"
},
"CI_upper": {
  "label": "CI_upper",
  "value": 80,
  "type": "integer"
},
"Percentile": {
  "label": "Percentile",
  "value": 14,
  "type": "integer"
},
"Repeat_spanning_coverage": {
  "label": "Repeat_spanning_coverage",
  "value": 21,
  "type": "integer"
},
"Qry_contig_length": {
  "label": "Qry_contig_length",
  "value": 249983.2,
  "type": "decimal"
},
"ImageText": {
  "label": "ImageText",
  "value": "Chromosome X, Map550 has a calculated repeat count of 2",
  "type": "string"
}
```



```
    }  
  }  
],  
  "type": "section",  
  "label": "Detailed results"  
}  
}  
]
```

Technical Assistance

For technical assistance, contact Bionano Genomics Technical Support.

You can retrieve documentation on Bionano products, SDS's, certificates of analysis, frequently asked questions, and other related documents from the Support website or by request through e-mail and telephone.

Type	Contact
Email	support@bionanogenomics.com
Phone	Hours of Operation: Monday through Friday, 9:00 a.m. to 5:00 p.m., PST US: +1 (858) 888-7663
Website	www.bionanogenomics.com/support